MENÚ SALIR

UNIVERSIDAD DE EXTREMADURA

Departamento de Informática
Escuela Politécnica de Cáceres

TESIS DOCTORAL

MÉTRICA DE IDONEIDAD DE ONTOLOGÍAS

Autor: Adolfo Lozano Tello
Directora: Asunción Gómez Pérez

Febrero 2002

Edita: Universidad de Extremadura
Servicio de Publicaciones
c/ Pizarro, 8
Cáceres 10071
Correo e.: publicac@unex.es
http://www.pcid.es/public.htm

Dña. Asunción Gómez Pérez, Titular de Universidad del Departamento de Inteligencia Artificial de la Universidad Politécnica de Madrid

CERTIFICA

Que D. Adolfo Lozano Tello, Licenciado en Informática, ha realizado en el Departamento de Informática de la Universidad de Extremadura, bajo mi dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada

Métrica de Idoneidad de Ontologías

Revisado el presente trabajo, estimo que puede ser presentado al tribunal que ha de juzgarlo y autorizo la presentación de esta Tesis Doctoral en la Universidad de Extremadura.

Cáceres, a 11 de febrero de 2002

Fdo. Dña. Asunción Gómez Pérez

Titular de Universidad
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid

MENÚ SALIR

A la memoria de mi padre

MENÚ SALIR

Agradecimientos

Difícilmente nadie podrá decir que el desarrollo de una Tesis sea posible sin los conocimientos científicos y el apoyo sentimental de la gente que te rodea. En mi caso esto tampoco es una excepción, por lo que debo agradecer infinitamente a mis compañeros y seres queridos su esfuerzo por hacer que este trabajo pueda finalizarse.

En primer lugar, debo agradecer a mi directora, Asunción Gómez, por haber puesto su confianza en mí. Además de haber sido un indiscutible referente científico y una revisora infatigable, para mí siempre será un modelo a seguir de voluntad, espíritu de trabajo y afán de superación.

No creo que pueda encontrar las palabras adecuadas para expresar la gratitud que debo a Mariano Fernández y Oscar Corcho, que me han ayudado a resolver tantas dudas teóricas y técnicas que difícilmente podría enumerar, y que siempre me han ofrecido gustosamente sus conocimientos. En sus áreas de conocimientos no creo que deba calificárseles como expertos, sino como sabios.

Muchas gracias a Miguel Ángel Vivas y Rocío Rodríguez por haber colaborado en el desarrollo de las aplicaciones software que se han necesitado para este trabajo, sé que les he pedido mucho más de lo que estaba previsto. También debo agradecer a la gente del *Grupo de Reutilización de Conocimientos* de la *UPM* por sus aportaciones a esta Tesis, en especial a Socorro, Raúl, Vanesa y José Ángel, y de *iSOCO* a Mercedes y Juan Manuel por haber sufrido todos el cuestionario.

A los compañeros del Departamento de Informática de la Universidad de Extremadura, por animarme y ayudarme a acabar esta Tesis, especialmente a Alberto Gómez que, además de amigo, ha sido un minucioso revisor.

Y por supuesto a mi madre, a mi hermano, y amigos por darme su apoyo moral para poder afrontar esta empresa. En especial, esta Tesis se la dedico de corazón a Gelu, mi compañera que ha sabido aconsejarme y soportarme en todo momento.

Muchas gracias a todos,

--Adolfo.

MENÚ SALIR

Resumen

En los últimos años, la creación de sistemas software basados en ontologías está aumentando considerablemente. Una ontología se define como una especificación explícita y formal de una conceptualización consensuada. Actualmente, los usuarios que buscan ontologías para incorporarlas a sus sistemas se basan únicamente en su experiencia e intuición, y esto hace difícil que puedan justificar las elecciones tomadas. Esto es debido, principalmente, a que no existe ningún método que indique al usuario qué ontologías son las más apropiadas para un nuevo sistema. Para resolver esta carencia, en este trabajo se propone el método ***OntoMetric***, que permite a los usuarios medir la idoneidad de las ontologías existentes respecto a las necesidades de sus sistemas. Sus principales aportaciones son:

- **Identificación de un marco multinivel de características**. El marco consiste en un conjunto de 160 características que describen el dominio de las ontologías. Se clasifican en las siguientes dimensiones: el contenido representado en la ontología, el lenguaje en el que se encuentra implementada, la metodología que se ha seguido para su construcción, los entornos software de desarrollo de ontologías, y los costes de usar las ontologías en el sistema. Este marco (a) proporciona el esquema para representar la información de ontologías existentes, (b) permite comparar las ontologías y (c) elegir las más apropiadas según las necesidades del nuevo sistema.

- **Construcción del modelo conceptual de una ontología en el dominio de las ontologías, *Reference Ontology* (RO), que está basada en el marco multinivel de características**. El modelo conceptual de la RO se ha desarrollado siguiendo la metodología *METHONTOLOGY* y el entorno de desarrollo *WebODE*. La dimensión "contenido" de esta ontología se ha instanciado con información de ontologías procedentes de servidores de ontologías y disponibles en la *web*. Las demás dimensiones se han instanciado con información de publicaciones existentes y también analizando directamente los lenguajes, metodologías y entornos de desarrollo de ontologías.

- **Elaboración de un método que mide la idoneidad de un conjunto de ontologías candidatas que pueden ser incorporadas en un nuevo sistema.** El método utiliza: (a) el marco multinivel de características, (b) el modelo conceptual de la RO y sus instancias, y (c) una adaptación del método de las jerarquías analíticas. Para cada una de las ontologías candidatas, el método ofrece una medida cuantitativa de su idoneidad y sirve para decidir, de forma justificada, qué ontologías son las más adecuadas para el sistema que el usuario va a desarrollar.

- **Construcción del soporte tecnológico que asiste al método**. Las aplicaciones software construidas son: a) ***OntoWrappers***: extraen la información de ontologías existentes en servidores de ontologías y páginas *web*; b) ***Reference Ontology Instances Selector***: selecciona las ontologías que cumplen ciertos requisitos indicados por los usuarios; y c) ***OntoMetric Tool***: calcula, usando una adaptación del método de las jerarquías analíticas, los valores de idoneidad para las ontologías candidatas.

MENÚ SALIR

Abstract

In the last years, the development of ontology-based applications has increased considerably. An ontology is a formal, explicit specification of a shared conceptualization. Users currently looking for ontologies in order to incorporate them into their systems, just use their experience and intuition. This makes it difficult for them to justify their choices. Mainly, this is due to the lack of methods that help the user to measure which the most appropriate ontologies for the new system are. To solve this deficiency, the present work proposes a method, ***OntoMetric***, which allows the users to measure the suitability of the existent ontologies, regarding the requirements of their systems. Its main contributions are:

- **Identification of a multilevel framework of characteristics.** The framework consists of 160 characteristics that describe the ontology domain. They are classified in the following dimensions: the content represented in the ontology, the language in which the ontology is implemented, the methodology followed to develop it, the software environments used for building it, and the costs of using the ontology in the system. This framework (a) provides the outline to represent the information of existing ontologies, (b) allows to compare the ontologies and (c) to choose the most convenient according to the new system requirements.
- **Building the conceptual model of an ontology in the ontologies domain, the *Reference Ontology* (RO), based on the multilevel framework of characteristics.** The conceptual model of the RO was built following *METHONTOLOGY* methodology and *WebODE* platform. The dimension "content" of this ontology is instantiated with information coming from ontologies stored on ontology servers and available on the web. The other dimensions were instantiated taking information from previous works, as well as by directly analyzing the languages, methodologies, and tools.
- **Design of a method to measure the suitability of a set of candidate ontologies to be incorporated into a new system.** The method uses: (a) the multilevel framework of characteristics, (b) the conceptual model of the RO and its instances, and (c) an adaptation of the analytic hierarchy process. For every candidate ontology, the method gets a quantitative measure of its suitability. This is useful to decide, in a justified way, which ontologies are the most appropriate for the application that the user is developing.
- **Creation of the technological support to assist the method.** The following software applications were built: a) *OntoWrappers*, to extract the information of existing ontologies in ontology servers and web pages; b) *Reference Ontology Instances Selector*, to select ontologies that meet user's requirements; and c) *OntoMetric Tool*, to calculate (using a modification of the analytic hierarchy process) the measurement of suitability for the candidate ontologies.

MENÚ SALIR

Índice

CAPITULO 1. INTRODUCCIÓN 1
1.1. EL PROBLEMA DE LA ELECCIÓN DE ONTOLOGÍAS 1
1.2. PROPUESTA PARA MEDIR LA IDONEIDAD DE LAS ONTOLOGÍAS 4
1.3. ORGANIZACIÓN DE LA MEMORIA 5

CAPITULO 2. ESTADO DE LA CUESTIÓN 7
2.1. INTRODUCCIÓN 7
2.2. LA REUTILIZACIÓN DEL CONOCIMIENTO 7
2.3. ONTOLOGÍAS 8
2.3.1. Términos de las Ontologías 9
2.3.2. Clasificaciones de Ontologías 10
2.3.3. Ontologías Existentes y Librerías de Ontologías 12
2.3.4. Aplicaciones Basadas en Ontologías 13
2.4. MARCOS DE TRABAJO PARA CLASIFICAR ONTOLOGÍAS 16
2.4.1. Características Básicas de Diseño de Gruber 16
2.4.2. Características Básicas de Diseño de Uschold y Grüninger 17
2.4.3. Esquema de Noy y Hafner 17
2.4.4. Esquema de Hovy 19
2.4.5. Marco de Características de Uschold 21
2.4.6. Conclusiones sobre los Marcos de Trabajo Existentes para Clasificar Ontologías 22
2.5. CARACTERÍSTICAS IDENTIFICADAS EN LAS DIMENSIONES PRINCIPALES DE REUTILIZACIÓN DE ONTOLOGÍAS 23
2.5.1. Características para Determinar el Contenido de las Ontologías 24
2.5.1.1. Características sobre Aspectos Estructurales del Contenido 24
2.5.1.2. Características Usadas para Identificar Términos de Ontologías 25
2.5.1.3. Conclusiones sobre Características Relacionadas con el Contenido 26
2.5.2. Características que Describen los Lenguajes de Implementación de Ontologías 26
2.5.3. Características que Describen las Metodologías de Desarrollo de Ontologías 31
2.5.4. Características que Describen los Entornos de Desarrollo de Ontologías 35
2.5.5. Conclusiones sobre las Clasificaciones de Características de Ontologías 37
2.6. MÉTRICAS DE REUTILIZACIÓN 39
2.6.1. Métricas en la Ingeniería del Software 39
2.6.2. Métricas de Reutilización en la Ingeniería del Software 41
2.6.3. Métricas de Reutilización en la Ingeniería del Conocimiento 42
2.6.4. Métodos de Decisión Multicriterio. El Método de las Jerarquías Analíticas 43
2.6.5. Conclusiones sobre las Métricas de Reutilización 46
2.7. CONCLUSIONES SOBRE EL ESTADO DE LA CUESTIÓN. 47

CAPITULO 3. PLANTEAMIENTO 51
3.1. INTRODUCCIÓN 51
3.2. OBJETIVOS Y APORTACIONES DEL TRABAJO 51
3.3. METODOLOGÍA DE TRABAJO 53
3.4. HIPÓTESIS DE TRABAJO 56

CAPITULO 4. DESARROLLO DEL MÉTODO *ONTOMETRIC* 59
4.1. INTRODUCCIÓN 59
4.2. VISIÓN GENERAL DEL MÉTODO *ONTOMETRIC*. 59
4.3. SOPORTE TECNOLÓGICO PARA EL MÉTODO *ONTOMETRIC* 62
4.3.1. Aplicaciones para Representar el Modelo Conceptual y las Instancias de la *Reference Ontology* 62
4.3.2. Aplicaciones para Asistir en la Aplicación del Método *OntoMetric* 63
4.4. MARCO MULTINIVEL DE CARACTERÍSTICAS PARA DESCRIBIR ONTOLOGÍAS. . 64
4.4.1. Definición del Marco Multinivel de Características de Ontologías 64
4.4.1.1. Las Dimensiones del Marco Multinivel de Características 64
4.4.1.2. Árbol Multinivel de Características 65
4.4.1.3. Características Relacionadas con la Dimensión “Contenido de Ontologías“ 67
4.4.1.3.1. Características Descriptivas Relacionadas con la Dimensión “Contenido” 68
4.4.1.3.2. Características de la Dimensión “Contenido” Usadas para la Medición de Idoneidad 69
4.4.1.4. Características Relacionadas con la Dimensión “Lenguajes de Implementación de Ontologías” 75
4.4.1.4.1. Características Descriptivas Relacionadas con la Dimensión “Lenguaje” 76
4.4.1.4.2. Características de la Dimensión “Lenguaje” Usadas para la Medición de Idoneidad 77
4.4.1.5. Características Relacionadas con la Dimensión ”Metodologías de Desarrollo de Ontologías” 79
4.4.1.5.1. Características Descriptivas Relacionadas con la Dimensión “Metodología” 80
4.4.1.5.2. Características de la Dimensión “Metodología” Usadas para la Medición de Idoneidad 80
4.4.1.6. Características Relacionadas con la Dimensión “Entornos de Desarrollo de Ontologías” 82
4.4.1.6.1. Características Descriptivas Relacionadas con la Dimensión “Entorno” 83
4.4.1.6.2. Características de la Dimensión “Entorno” Usadas para la Medición de Idoneidad 84
4.4.1.7. Características Relacionadas con la Dimensión “Costes de Uso de Ontologías” 92
4.4.2. Representación del Marco Multinivel de Características en una Ontología: *La Reference Ontology* 93
4.5. INSTANCIACIÓN DE LA *REFERENCE ONTOLOGY* 104
4.5.1. Instanciación de la Dimensión “Contenido” de Ontologías 106

4.5.2. Instanciación de las Dimensiones: “Lenguaje”, “Metodología”, “Entorno” y “Costes” 108
4.6. SELECCIÓN DE ONTOLOGÍAS CANDIDATAS 116
4.6.1. Herramienta Software para la Selección de Ontologías: *Reference Ontology Instances Selector* 116
4.7. MEDICIÓN DE IDONEIDAD DE ONTOLOGÍAS CANDIDATAS. 118
4.7.1. El Método de las Jerarquías Analíticas Adaptado a la Elección de Ontologías . 118
4.7.1.1. Paso 2: Organización del Árbol Jerárquico de Criterios 119
4.7.1.2. Paso 3: Ponderación de los Criterios del Árbol Multinivel de Características 121
4.7.1.2.1. Obtención de los Pesos de Importancia a partir de las Matrices de Comparación de Criterios 121
4.7.1.2.2. Pesos por Defecto Asignados al Árbol Multinivel de Características 123
4.7.1.3. Paso 4: Valoración de los Nodos Hoja del Árbol Multinivel de Características para las Ontologías Candidatas 127
4.7.1.3.1. Valores Asociados a Escalas Lingüísticas 127
4.7.1.3.2. Asignación de Valores a las Criterios del AMCP a partir de las Instancias de la *Reference Ontology* 129
4.7.1.4. Paso 5: Cálculo y Comparación de los Valores de Idoneidad para las Ontologías Candidatas 130
4.7.1.5. Diferencias de OntoMetric con el Método de las Jerarquías Analíticas 133
4.7.2. Herramienta Software para Asistir a la Aplicación del Método: *OntoMetric Tool*. 133

CAPITULO 5. EVALUACIÓN DEL MÉTODO *ONTOMETRIC* CON CASOS DE ESTUDIO 135
5.1. INTRODUCCIÓN 135
5.2. CASOS DE PRUEBA PARA EVALUAR CADA DIMENSIÓN INDIVIDUALMENTE 138
5.2.1. Casos de Prueba para la Dimensión “Lenguajes de Implementación de Ontologías” 139
5.2.2. Casos de Prueba para la Dimensión “Metodologías de Desarrollo de Ontologías” 142
5.2.3. Casos de Prueba para la Dimensión “Entornos Software de Desarrollo de Ontologías” 145
5.2.4. Casos de Prueba sobre la Dimensión “Contenido de Ontologías” 147
5.3. CASOS DE PRUEBA PARA EVALUAR LAS DIMENSIONES DE FORMA CONJUNTA 155
5.4. CONCLUSIONES DE LA EVALUACIÓN DEL MÉTODO *ONTOMETRIC* 170

CAPITULO 6. CONCLUSIONES Y LÍNEAS FUTURAS. 173

BIBLIOGRAFÍA. 177

ANEXOS:

I. Cuestionario de Opinión sobre Características para Seleccionar Ontologías
II. Descripción del Funcionamiento de los *OntoWrappers*
III. Cálculo de Pesos por Defecto a partir de los Cuestionarios de Opinión

MENÚ SALIR

Capítulo 1

Introducción

> "Cuando puedas medir lo que estás diciendo y expresarlo con números, ya conoces algo sobre ello; cuando no puedas medir, expresando lo que dices con números, tu conocimiento es precario y deficiente; puede ser el comienzo del conocimiento, pero en tus pensamientos apenas estás avanzando en el escenario de la ciencia." (Lord Kelvin, 1824-1907)

1.1. EL PROBLEMA DE LA ELECCIÓN DE ONTOLOGÍAS

En 1991, *the ARPA Knowledge Sharing Effort* [Nec91] revolucionó la forma de construir sistemas inteligentes en Inteligencia Artificial al proponer la construcción de sistemas basados en conocimientos mediante el "ensamblaje" de componentes reutilizables. Dichos componentes reutilizables se convierten en la base (o esqueleto) del nuevo sistema, al cual se le añaden conocimientos especializados y métodos de razonamiento específicos, propios de la tarea que el sistema pretende resolver. Además, los nuevos sistemas interoperan con los sistemas existentes, usando parte de sus métodos de razonamiento. Así, los conocimientos declarativos del dominio, las técnicas de resolución de problemas y los razonamientos que cada sistema posee se comparten con otros sistemas. Este enfoque permite construir sistemas mayores y más potentes. Surge así el área relacionada con la reutilización y compartición de componentes basados en conocimientos.

Las ontologías, usadas para representar los conocimientos estáticos de un dominio y los métodos de resolución de problemas, en inglés *problem solving methods*, utilizados para realizar razonamientos, se convierten así en las piezas claves que permiten la reutilización de conocimientos y métodos de razonamiento. Una ontología se define como una especificación explícita y formal de una conceptualización consensuada [Stu98]. Los métodos de resolución de problemas describen los procesos de razonamiento de un sistema basado en conocimientos de una manera independiente de la implementación e independiente del dominio [Gom99a]. La integración de ontologías y métodos de resolución de problemas es una de las mejores soluciones [Gom99a] al problema de la interacción [Byl88], que postula que la representación de los conocimientos del dominio está muy condicionada por el tipo de problema y por las técnicas de razonamiento que se utilizan para resolver este problema.

En la década de los 90 se desarrollaron numerosas ontologías, las cuales están disponibles en servidores de ontologías y en la *web*. Algunas de las más representativas son: *CYC* [Len90], *WordNet* [Mil90], *Frame Ontology* [Gru93b], *PANGLOSS* [Kni94], *TOVE* [Gru95b], *Mikrokosmos* [Mah96], *Enterprise ontology* [Usc96a], *CHEMICALS* [Fer96], *Mereology* [Bor97], *SENSUS* [Swa97], *(KA)^2 ontology* [Ben99], etc.

Actualmente, las ontologías se encuentran implementadas en una gran variedad de lenguajes. Al principio de la década de los 90, un conjunto de lenguajes fueron diseñados y utilizados para la implementación de ontologías. Los lenguajes más representativos son: *Ontolingua* [Gru93b], *LOOM* [MGr91], *OCML* [Mot99], *FLogic* [Kif95], etc. Estos lenguajes reciben el nombre de "lenguajes clásicos" [Cor00], siguen una sintaxis basada en *LISP* (a excepción de *FLogic*), y están en una fase de desarrollo estable. Recientemente, *XML* ha sido adoptado como un lenguaje estándar para intercambiar información en la *web*, y en el campo de las ontologías se han creado lenguajes basados en *XML* para implementar ontologías. Por ejemplo *RDF* [*Las99*], *RDF Schema* [Bri99], *SHOE* [Luk00], *XOL* [Kar99], *OIL* [Hor00] y *DAML+OIL* [Hor01]. Estos lenguajes, llamados "lenguajes de marcado *web*", están aún en fase de desarrollo y en continua evolución.

Igualmente, han sido numerosas las metodologías de desarrollo de ontologías propuestas. Ya en 1990, Lenat y Guha [Len90] publicaron algunas consideraciones metodológicas relacionadas con el desarrollo de la ontología *CYC*. Algunos años más tarde, en 1995, Uschold y King [Usc95] publicaron los principales pasos que siguieron en el desarrollo de la ontología *Enterprise*. En el mismo año, Grüninger y Fox [Gru95b] mostraron la metodología utilizada en el desarrollo de la ontología *TOVE* (*TOronto Virtual Enterprise*). Un año más tarde, Uschold [Usc96a] realiza una propuesta de unificación de ambas metodologías. En la *12th European Conference for Artificial Intelligence* se presenta la metodología utilizada para construir ontologías dentro del proyecto *Esprit KACTUS project* [Ber96]. En 1997 aparece *METHONTOLOGY* [Fer97], la cual fue extendida posteriormente en [Fer99a] [Fer00], y en la que se incluyen, además de los pasos que se deben seguir para construir ontologías, unas guías para realizar reingeniería de ontologías [Gom99d] y evaluación de ontologías [Gom99c]. También en 1997 se propuso [Swa97] la metodología empleada para construir ontologías de dominios a partir de la ontología *SENSUS*. Todas estas metodologías no consideran el desarrollo colaborativo de ontologías. La primera metodología que incluye aspectos de desarrollo en grupo y colaborativo es *Co4* [Euz95]. Un estudio comparativo de algunas de estas metodologías aparece en [Fer99b].

La década de los 90 también supuso un importante avance en el desarrollo de plataformas tecnológicas relacionadas con las ontologías. El primer servidor de ontologías que apareció fue el *Ontolingua Server* [Far96], del *Knowledge Systems Laboratory* (*KSL*) en la *Stanford*

University. En 1997, apareció *Ontosaurus* [Swa97], desarrollado por el *Information Sciences Institute* (*ISI*) en la *University of South California*. Posteriormente, se han creado herramientas basadas en tecnología *Java*: *WebOnto* [Dom98] desarrollada en el *Knowledge Media Institute* (*KMI*) de la *Open University* (*UK*); *OILed* [Bec01], desarrollada en el contexto del proyecto europeo IST *OntoKnowledge project*; *OntoEdit* [Sta00] desarrollada por el *AIFB* de la *Karlsrhue University*; *Protégé2000* [Noy01] desarrollado por el *Stanford Medical Informatics* (*SMI*) en la *Stanford University*; y *WebODE* [Arp01] desarrollada en la Facultad de Informática de la Universidad Politécnica de Madrid.

Actualmente, los ingenieros del conocimiento encargados de desarrollar sistemas basados en ontologías buscan en la red, o en sus servidores conocidos, las ontologías del dominio que le pueden servir para su nuevo sistema. Cuando encuentran varias que pueden ser adecuadas, deben examinar detenidamente sus características y decidir cuáles son las mejores para incorporarlas a su sistema. Este procedimiento es totalmente artesanal, y la elección suele depender, sobre todo, de la experiencia e intuición del ingeniero. Si el sistema está siendo desarrollado con fines comerciales, le será muy difícil justificar la elección tomada.

Como se ha dicho anteriormente, son muchas las ontologías existentes, y numerosos los lenguajes en los que se pueden encontrar implementadas. Una misma ontología se puede encontrar implementada en diferentes lenguajes y, dependiendo de esto, ciertas implementaciones pueden ser más apropiadas para ciertos sistemas (o usos) que otras. Por este motivo, a medida que aumentan las ontologías, los entornos y metodologías que se usan para desarrollarlas, y los lenguajes de implementación, cada vez va a ser más difícil decidir qué ontologías son más apropiadas para una nueva aplicación.

Aunque la mayoría de las metodologías de desarrollo de ontologías [Fer99b] proponen como una de sus fases la reutilización de ontologías existentes, no se ha encontrado ningún trabajo que indique al usuario cómo elegir ontologías para el proyecto que va a desarrollar ni que cuantifique la adecuación de estas ontologías para el sistema.

Este problema de elección se paliaría si existiera una métrica que cuantificara, para cada una de las ontologías candidatas, cómo de adecuadas son para un nuevo sistema. El método que se describe en este trabajo (denominado ***OntoMetric***) presenta el conjunto de procesos que debe realizar el usuario para obtener estas medidas.

1.2. PROPUESTA PARA MEDIR LA IDONEIDAD DE LAS ONTOLOGÍAS

Según la definición de Fenton [Fen96], "una medición es el proceso por el cual números o símbolos se asignan a atributos de entidades del mundo real, de forma que lo describen de acuerdo a reglas claramente definidas". La métrica que propone *OntoMetric* indica cómo asignar un valor a una ontología y, para ello, detalla los procesos que deben realizarse para obtener esta medida. El atributo de la ontología que va a ser medido es su idoneidad en relación con las necesidades de un proyecto determinado. En el estándar ISO 9126:1991 A.2.1.1 [ISO91], se define el concepto de **idoneidad** del software como "los atributos del software que se refieren a la presencia y adecuación de un conjunto de características para realizar tareas específicas". Basándose en esta definición, el concepto de idoneidad de una ontología que se va a seguir en este trabajo se identifica con las propiedades de la ontología consideradas en conjunto para determinar su adecuación en la representación de los conocimientos necesitados en un sistema determinado.

La medida de idoneidad que se obtenga va a depender directamente de las metas del proyecto, por lo que antes de aplicar la métrica deberán estar perfectamente especificados los objetivos y pretensiones del proyecto. Esta es una hipótesis básica y fundamental para que el método *OntoMetric* pueda aplicarse. Referenciando el principio de Gilb [Gil88] (y juego de palabras [1]) sobre los objetivos difusos: "Los proyectos sin metas claras, es obvio que no alcanzarán sus metas". Por ello, si el usuario que va a aplicar *OntoMetric* no conoce de forma precisa los objetivos de su proyecto o no sabe la importancia que tiene cada uno de ellos, no existe garantía en los resultados obtenidos.

El método *OntoMetric* ha adaptado el método de las jerarquías analíticas [Saa77] para la reutilización de ontologías. Los criterios básicos de decisión, denominados dimensiones, son los aspectos fundamentales que un usuario debe considerar antes de elegir una ontología: debe examinar en qué grado el **contenido** de la ontología y su organización se ajustan a las necesidades del proyecto; si el **lenguaje** en el que está implementada la ontología es el adecuado para ser incorporado al sistema; si la **metodología** seguida para desarrollar la ontología permite garantizar la fiabilidad de su contenido, y si permite realizar fácilmente procesos de adaptación; los **entornos software** capaces de visualizar, editar y explotar el contenido de la ontología; y la previsión de los **costes** que acarreará usar la ontología elegida. Cada una de estas dimensiones contiene una serie de factores que servirán para determinar la idoneidad de cada una de ellas a las necesidades del proyecto, y cada factor, a su vez, una serie

[1] Frase original del principio de Gilb: "*Projects without clear goals will not achieve their goals clearly*".

de características que determinarán su valor, pudiéndose especificar éstas mediante otras subcaracterísticas más específicas y así, de forma recursiva, establecer el grado de detalle con el que se quiere precisar el análisis de la ontología.

Como se ha mencionado, antes de desarrollar un proyecto software deben estar especificados de forma precisa el conjunto de objetivos que se quieren alcanzar. Así, para escoger una o varias ontologías e integrarlas en su proyecto, el ingeniero debe tener en cuenta conjuntamente todos los objetivos y la importancia que tiene cada uno de ellos. Este problema se puede incluir en los problemas de decisión multicriterio [Vin92], en los que, más que resolver el problema con un resultado demostrado por datos, se encuentra una solución subjetiva al problema, que es consistente y satisface de la mejor forma a las partes involucradas. Los métodos que resuelven estos problemas proporcionan un sistema para organizar juicios de apreciación, que los "decisores" usan en los procesos de elección donde existan varias alternativas. El método de las jerarquías analíticas [Saa77] (*Analythic Hierarchy Process*), en el que se basa este trabajo, es uno de los más populares y usados métodos de decisión multicriterio.

El resultado final de este proceso es una valoración de idoneidad para cada una de las ontologías que se examinen. Esto ayudará al usuario a tomar una decisión justificada, según la importancia que le haya dado a cada uno de los criterios y de acuerdo con los valores que tome cada ontología en cada una de las características identificadas.

1.3. ORGANIZACIÓN DE LA MEMORIA

Esta memoria se ha organizado en los siguientes capítulos:

2. Estado de la cuestión: Se recopilan las publicaciones y trabajos existentes relacionados con ontologías y con las métricas de reutilización en Ingeniería de Software e Ingeniería del Conocimiento. Respecto a las ontologías, se ha hecho un detallado estudio de las publicaciones que aportaban características en las citadas dimensiones.
3. Planteamiento: Se han identificado las principales carencias y dificultades en el proceso de reutilizar ontologías, y se han establecido las tareas que deben desarrollarse para resolver estos problemas. Se indican los objetivos, principales aportaciones, y las hipótesis de trabajo.
4. Desarrollo del método *OntoMetric*: Se detallan las tareas que se han seguido para desarrollar el método *OntoMetric* y se describen los procesos que deben realizar los usuarios que quieran utilizar el método. En los procesos que correspondan, se describe el software que ayuda a aplicar el método.

5. Evaluación del método *OntoMetric*: Se exponen algunos de los casos de prueba realizados que han servido para evaluar la aplicabilidad del método.
6. Conclusiones y líneas futuras: Se resumen las aportaciones de este trabajo y se describen posibles trabajos futuros que pueden acometerse a partir de la situación actual.

Se finaliza con la bibliografía y los anexos.

Capítulo 2

Estado de la Cuestión

2.1. INTRODUCCIÓN

En este capítulo se recogen los trabajos existentes relacionados, principalmente, con ontologías y métricas de reutilización. Respecto a las ontologías, se definen los conceptos fundamentales y se resume el estado del arte en esta área. Se ha realizado un estudio detallado de los trabajos que definen características relacionadas con la reutilización de ontologías, desde los marcos de trabajo existentes para comparar ontologías, hasta los estudios específicos de algún aspecto concreto sobre ontologías: lenguajes de implementación de ontologías, metodologías de desarrollo, entornos software de desarrollo, y trabajos que analizan el contenido de las ontologías. Respecto a las métricas de reutilización, se han resumido las métricas existentes relacionadas con la reutilización de componentes en Ingeniería del Software y la reutilización de conocimientos en Ingeniería del Conocimiento. Se describe también en este capítulo el método de las jerarquías analíticas, por ser el fundamento teórico en el que se va a basar el método propuesto en este trabajo.

2.2. LA REUTILIZACIÓN DEL CONOCIMIENTO

Los procesos de reutilización de componentes software y componentes basados en conocimientos abarcan tanto al desarrollo de sistemas de Ingeniería del Software clásico como a los de Ingeniería del Conocimiento. El *National Institute of Standards and Technology* [NIS94] define los *assets*, o componentes reutilizables, como "cualquier producto del ciclo de vida del software que pueda ser potencialmente reutilizado. Esto incluye: modelos de dominios, arquitecturas de dominios, requisitos, diseño, código, bases de datos, esquemas de bases de datos, documentación, manuales de usuario, casos de prueba,...". Así, entre los componentes software reutilizables se encuentran los componentes basados en conocimientos (entre los que están ontologías y métodos de resolución de problemas [Gom99a]). El ahorro en costes y tiempo que se obtiene en la reutilización del software [Bol90][Pou97] se logra en mayor medida en la reutilización de conocimientos, debido al enorme esfuerzo que conllevan los procesos de

adquisición de conocimientos de un dominio, la construcción de su modelo conceptual, y la formalización e implementación de tales conocimientos.

En 1991, Neches y colegas [Nec91] propusieron como solución a estos problemas la construcción de sistemas basados en conocimientos (SS.BB.CC.) uniendo o "ensamblando" componentes reutilizables, de manera que los desarrolladores de los nuevos sistemas sólo tuvieran que preocuparse de la creación de conocimientos especializados y de desarrollar nuevos razonadores para la tarea específica de su sistema. Los nuevos sistemas interoperarían con los sistemas existentes, compartiendo así sus conocimientos y métodos de razonamiento. El conocimiento declarativo incluido en ontologías, las técnicas de resolución de problemas y los métodos de razonamiento podrían estar en los sistemas compartidos, proporcionando sistemas mejores, mayores y más baratos de construir. El objetivo era construir sistemas ensamblando conocimientos y métodos de razonamiento, en vez de desarrollarlos desde la nada.

2.3. ONTOLOGÍAS

En filosofía, el término ontología se define como "la parte de la metafísica que trata del ser en general y de sus propiedades trascendentales" [RAE99]; en Inteligencia Artificial, en cambio, tiene diferentes connotaciones. Existen diferentes autores que ofrecen su propia interpretación de este concepto, algunas recogidas y analizadas en 1995 por Guarino y Giaretta [Gua95]. La definición declarativa más consolidada es la propuesta por Gruber [Gru95a], y extendida por Studer y colegas [Stu98], como "una especificación explícita y formal sobre una conceptualización consensuada". La interpretación de esta definición es que las ontologías definen sus conceptos, propiedades, relaciones, funciones, restricciones y axiomas de forma "explícita" en algún lenguaje de implementación capaz de contener este conocimiento. El término "conceptualización" se refiere a un modelo abstracto de algún fenómeno en el mundo. El conocimiento de las ontologías es establecido para ser usado de forma "consensuada" y compartida por diferentes sistemas, que deberán comprometerse con el vocabulario utilizado en la ontología. El término "formal" se refiere a que la ontología debe implementarse en algún lenguaje computable por la máquina.

Desde la perspectiva de las aplicaciones que las utilizan, existen tres definiciones diferentes de ontología, aunque complementarias:

- Para Swartout y colegas [Swa97], "una ontología es un conjunto de términos estructurados jerárquicamente que describen un dominio. La ontología será el "esqueleto" sobre el cual se construye luego la base de conocimientos". La característica principal que se acentúa en esta definición es que las ontologías proporcionan la estructura taxonómica de un dominio,

la cuál será especializada con los conocimientos específicos necesitados por la aplicación.

- Para Bernaras y colegas [Ber96], "una ontología proporciona los significados que describen explícitamente la conceptualización del conocimiento representado en una base de conocimientos". Mediante un proceso de abstracción, el modelo conceptual de la ontología se construye a partir de los conocimientos procedentes de bases de conocimientos existentes. De esta forma, los conocimientos especificados en la ontología pueden ser utilizados más fácilmente para construir otras bases de conocimientos en el mismo dominio, o en dominios similares.
- Para Mizoguchi y colegas [Miz93], una ontología es "un sistema de conceptos/vocabulario usados como primitivas para construir sistemas artificiales". En este sentido, el propósito en la construcción de ontologías es capturar los conocimientos para que puedan ser empleados en diferentes sistemas, independientemente de la tarea que pretendan resolver.

A modo de conclusión, estas tres definiciones tienen en común la idea de que las ontologías proporcionan la conceptualización explícita de los términos de un dominio, que sirve como soporte para la implementación de bases de conocimientos preparadas para ser utilizadas por aplicaciones, y que resuelven diferentes tareas. No obstante, además de ser la base para los sistemas basados en conocimientos, actualmente el ámbito de aplicación de las ontologías abarca también otras áreas: procesamiento de lenguaje natural, *web* semántica, comercio electrónico, gestión de conocimientos, etc.

2.3.1. Términos de las Ontologías

Las ontologías deben ser creadas conforme a unos determinados criterios de diseño [Gru95a][Gua00] y, para que este campo sea considerado como una ingeniería "madura", las ontologías deben ser desarrolladas con adecuadas metodologías de desarrollo [Fer99b]. Según Uschold y Grüninger [Usc96b], dependiendo del propósito para el que sea creada, será construida con distintos grados de formalidad, desde los altamente informales expresados en lenguaje natural (lo cual contradice la definición de Gruber [Gru95a] y extendida posteriormente por Studer y colegas [Stu98] que indican que las ontologías deben ser especificadas formalmente), hasta los rigurosamente formales expresados en lenguajes con lógica de primer orden. Así, la apariencia de una ontología altamente formal será muy distinta a otra cuyas definiciones estén expresadas en lenguaje natural. En la mayoría de las ontologías aparecerán los siguientes términos [Gru93b][Gom98]:

- **Conceptos** o clases: son colecciones de objetos del dominio.
- **Relaciones**: representan interacciones entre conceptos del dominio. Suelen aparecer las

relaciones del tipo: *subclase-de*, *parte-de*, *conectado-a*, etc. Pueden ser binarias o n-arias dependiendo de los argumentos que tomen.

- **Funciones**: son un tipo especial de relación en las que, para un conjunto de argumentos de entrada, se obtiene un único argumento de salida. Por ejemplo, pueden aparecer funciones como *días-de-demora*, *precio-de-coche*, etc.
- **Instancias**: representan elementos determinados de una clase o concepto.
- **Axiomas**: son fórmulas, en la mayoría de las ocasiones expresadas en lógicas de primer orden, que se usan para modelar sentencias que son siempre verdad, y se utilizan para verificar la corrección de la información contenida en la ontología. Los axiomas también son usados para realizar inferencias sobre los conocimientos almacenados.

En función de estos términos que se utilizan para representar los conocimientos del dominio, se distinguen dos tipos de ontologías: las ontologías ligeras (*lightweight ontologies*) y las ontologías pesadas (*heavyweight ontologies*) [Gom99b]. Las ontologías ligeras contienen todos los términos excepto los axiomas. Las ontologías pesadas sí contienen axiomas, además de los demás elementos, y cuantos más axiomas contengan serán consideradas más "pesadas".

2.3.2. Clasificaciones de Ontologías

Existen diferentes clasificaciones de ontologías dependiendo del punto de vista que se quiera considerar. Mizoguchi y colegas [Miz95] clasifican las ontologías desde la perspectiva del tipo de problemas que resuelven. En el nivel superior de su clasificación, como se muestra en la figura 2.1, consideran cuatro categorías:

- *Ontologías de contenido*: construidas para reutilizar su conocimiento, vistas como cajas blancas que permiten reutilizar su vocabulario.
- *Ontologías de indexación*: permiten la recuperación de casos cuando los agentes comparten conocimientos a través de bases de casos.
- *Ontologías de comunicación*: usadas por agentes para obtener respuestas a preguntas concretas, vistas como cajas negras de conocimientos.
- *Meta-ontologías*: utilizadas para representar ontologías [2].

En función del grado de reutilización de cada ontología, van Heist y colegas [vHe97] proponen la clasificación que aparece en la figura 2.1. Atendiendo al tema (o ámbito) de la conceptualización, clasifican las ontologías en:

[2] Este término empleado por Mizoguchi y colegas no es el más usual. El concepto de meta-ontología más aceptado [Gom99b] es el de ontologías genéricas, acorde con la definición de van Heist [vHe97].

- *Ontologías de representación*: proporcionan conceptualizaciones subyacentes a los paradigmas o formalismos de representación del conocimiento, es decir, proporcionan el vocabulario necesario para modelizar otras ontologías, utilizando un determinado paradigma de representación del conocimiento. El ejemplo más característico es la *Frame Ontology* [Far96] disponible en el servidor de *Ontolingua* [3], que proporciona el vocabulario necesario para representar una ontología siguiendo el paradigma de marcos.
- *Ontologías genéricas* o meta-ontologías: proporcionan términos genéricos reutilizables en diferentes dominios. Como, por ejemplo, los términos: estado, evento, acción, componente, etc.
- *Ontologías de dominio*: expresan conceptualizaciones que son específicas para dominios particulares. Los conceptos en las ontologías de dominio son usualmente definidos como especializaciones de conceptos existentes en ontologías genéricas.
- *Ontologías de aplicación*: contienen todas las definiciones que son necesarias para modelizar los conocimientos requeridos por una aplicación particular. Incluyen conceptos tomados de ontologías de dominio y genéricas, a menudo definidas utilizando el vocabulario indicado en las ontologías de representación. Pueden contener extensiones de métodos y tareas específicas. Las ontologías de aplicación tienden a ser menos reutilizables que las anteriores, ya que son especificaciones concretas del dominio que se

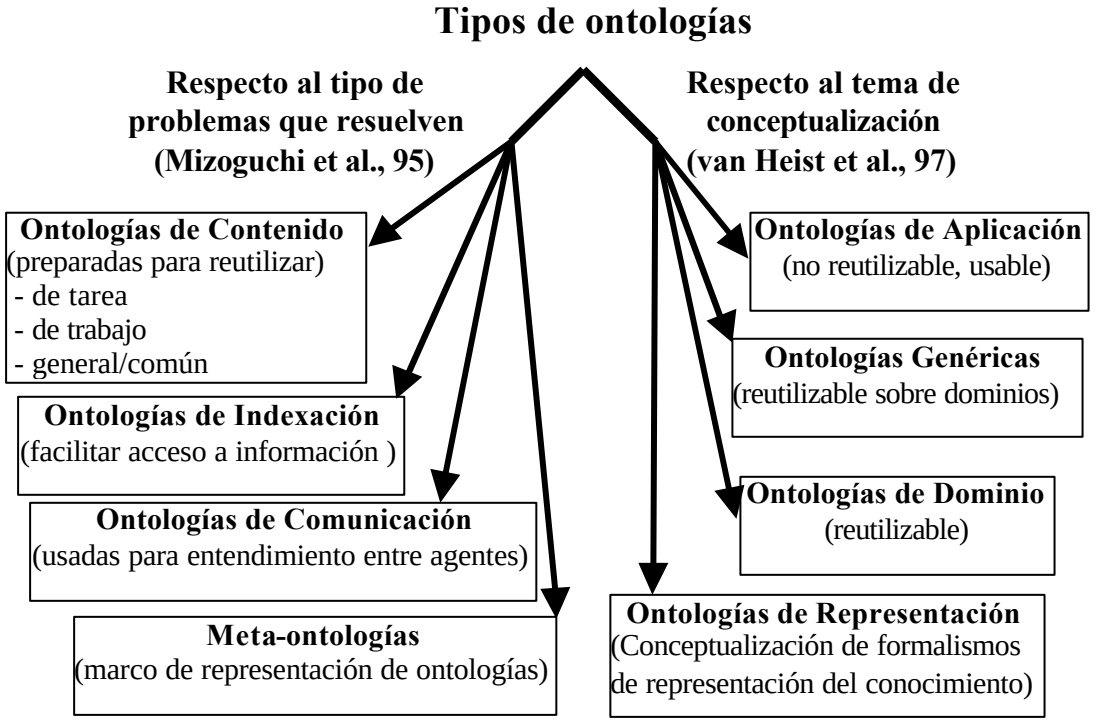


Figura 2.1. Clasificaciones de ontologías respecto al tipo de problemas que resuelven, y respecto al tema de conceptualización.

[3] http://www-ksl-svc.stanford.edu:5915

necesita para realizar una tarea particular en ese dominio determinado.

En ocasiones, es difícil clasificar una ontología sólo en uno de estos tipos, ya que el límite entre ellos no está definido claramente.

2.3.3. Ontologías Existentes y Librerías de Ontologías

En la actualidad existe un auge en el desarrollo de ontologías, tanto en ámbitos científicos como empresariales. Existen varias iniciativas que impulsan el uso de ontologías, como *Ontology.Org Library LinkBank* [4] para la promoción de aplicaciones basadas en ontologías, especialmente en el campo del comercio electrónico; y en los últimos años se están desarrollando muchas ideas relacionadas con la *web* semántica (*The Semantic Web* [5] [Ber01]), *OntoWeb* [6] y *DAML* [7].

Atendiendo a la clasificación de van Heist y colegas [vHe97], en este apartado se presenta una breve muestra de las ontologías más conocidas:

- Ontologías de representación: la más utilizada es la *Frame Ontology* [Far96] del *Ontolingua Server*, que contiene primitivas de representación usadas en lenguajes basados en marcos; está implementada en *KIF* [Gen92].
- Ontologías genéricas: el ejemplo más típico de ontologías genéricas es *Mereology* [Bor97] que incluye, entre otras, definiciones de la relación *parte_de* y de sus propiedades. Una de las ontologías más populares, dentro de esta clasificación, es *CYC* [8] [Len90], que contiene una enorme cantidad de términos de sentido común. Dentro de las ontologías genéricas se pueden referir las que contienen conceptos de nivel superior, como la ontología de Sowa [Sow97], *PANGLOSS* [Kni94], *Penman Upper Level* [Bat90], *Mikrokosmos* [Mah96] o la ontología de nivel superior de Guarino [Gua98]. Hay que destacar el esfuerzo que está realizando la comunidad científica para desarrollar una ontología llamada SUO [9] (*Standard Upper Ontology*) [Nil01]. Se pretende que los términos de esta ontología sirvan como el nivel superior de una taxonomía de términos consensuada por la comunidad que trabaja en el área de las ontologías, bajo la cual situar los términos de ontologías más específicas.
- Ontologías de dominio: continuamente se publican ontologías en muchos dominios; se señala aquí sólo una muestra representativa bien conocida por la comunidad que trabaja en

[4] http://www.linkbank.net/
[5] http://www.SemanticWeb.org/ontologies/
[6] http://www.ontoweb.org
[7] http://www.daml.org
[8] http://www.cyc.com/
[9] http://suo.ieee.org/

el área de las ontologías. En el dominio de la ingeniería se puede mencionar *EngMath* [Gru94] y *PhysSys* [Bor96]. En el dominio del proceso de modelado empresarial destacan *Enterprise Ontology* [10] [Usc96b] y varias ontologías del proyecto *TOVE* [11] [Gru95b]. En el dominio médico *UMLS* [12] [Hum93]. Entre otras ontologías lingüísticas, aunque muchas de ellas también podrían ser consideradas como genéricas, se pueden citar *Wordnet* [13] [Mil90], *GUM* [14] [Bat95] y *SENSUS* [Swa97]. Un ejemplo ilustrativo de gestión del conocimiento es la ontología *(KA)²* [Ben99].

- Ontologías de aplicación: se pueden examinar las continuas noticias publicadas en *Ontology.Org Library LinkBank* para comprobar cómo se construyen y se adaptan ontologías para aplicaciones concretas.

La mayoría de las ontologías existentes se encuentran almacenadas en repositorios, como las ontologías del servidor de *Ontolingua* en el *Ontolingua Server* [Far96]; en las páginas *web* de *SHOE* [15] [Hef00]; del *Ontology Group at ITBM-CNR* [16] del proyecto *ON9* [Ste00] con ontologías en el dominio médico; y las ontologías accesibles desde *DAML Ontology Library* [17] implementadas en *DAML+OIL*, cuyo lenguaje de marcado *web* se está convirtiendo en uno de los más usados para la representación de ontologías utilizadas en aplicaciones *web*. Existen otras librerías de ontologías que no son de dominio público, accesibles de forma restringida mediante entornos *web*, como son *WebODE* [18] [Arp01], *WebONTO* [19] [Dom98] y, parcialmente, *Ontosaurus* [20] [Swa97].

2.3.4. Aplicaciones Basadas en Ontologías

Las ontologías son usadas en diversos proyectos de investigación y empresariales, en áreas como el procesamiento de lenguaje natural, sistemas de información cooperativos, integración de información inteligente, comercio electrónico, etc., ya que proporcionan una visión compartida y común de un dominio, entre las personas y las aplicaciones [Fen01].

Referente a cómo las ontologías son usadas en diferentes sistemas y entornos, Uschold y

[10] http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html
[11] http://www.ie.utoronto.ca/EIL
[12] http://wwwkss.nlm.nih.gov/Docs/umls.fact.html
[13] http://www.tio.darpa.mil/Summaries95/B370-Pricenton.html
[14] http://www.darmstadt.gmd.de/publish/komet/gen-um/newUM.html
[15] http://www.cs.umd.edu/projects/plus/SHOE/onts/
[16] http://saussure.irmkant.rm.cnr.it/onto/
[17] http://www.daml.org/ontologies/
[18] http://babage.dia.fi.upm.es/webode/
[19] http://kmi.open.ac.uk/projects/weboto/
[20] http://www.isi.edu/isd/ontosaurus.html

Jasper [Usc99] proponen una clasificación de las aplicaciones que usan ontologías con algún fin práctico. Su clasificación se basa en un conjunto de escenarios, tomados de la industria o de entornos académicos, donde las ontologías pueden ser aplicadas para alcanzar uno o más propósitos. Cada escenario se caracteriza por: el objetivo que se pretende conseguir, el papel que desempeña la ontología, los actores que se necesitan para implementar el escenario, la tecnología utilizada y su nivel de madurez. Una misma aplicación basada en ontologías puede pertenecer a más de un escenario de los propuestos. Los autores categorizan estos escenarios en tres grandes áreas:

- *Realización neutral*: una ontología sobre un dominio determinado es implementada en un único lenguaje. Diferentes aplicaciones usarán la ontología como fuente de conocimientos, pero traducida al lenguaje que necesite la aplicación. Los beneficios de esta aproximación son la reutilización del conocimiento y el almacenamiento de gran cantidad de términos.
- *Acceso común a información*: En algunos casos, se necesita algún tipo de información para una o más personas y aplicaciones con un vocabulario no familiar para todos, o con formatos inaccesibles. La ontología se construye proporcionando un vocabulario común y consensuado para las múltiples aplicaciones, permitiendo el acceso a una fuente de información homogénea y comprensible. La diferencia con el escenario anterior es que, en este caso, no se traduce a diferentes lenguajes o formatos, sino que la ontología se mantiene como estándar común que deben usar las diferentes personas y aplicaciones. Los beneficios que se logran son la interoperatividad y un uso más efectivo de las fuentes de conocimientos.
- *Indexación*: aplicaciones que realizan búsquedas basadas en conceptos (por ejemplo documentos en algún repositorio) utilizan una ontología como índice para localizar términos. Se obtiene un acceso más rápido, consensuado y controlado a los recursos de información almacenados.

Aunque el objetivo de crear ontologías es poder compartir y reutilizar su conocimiento en otros sistemas, aún no se ha conseguido que estos procesos se realicen de forma sistemática. Esto es debido a diversos problemas [Arp00a]: las ontologías están diseminadas en diferentes servidores, la representación difiere dependiendo del servidor en el que están almacenadas, suelen estar descritas con diferente nivel de detalle, y no existe un formato común para describir toda la información relevante donde el potencial usuario pueda examinar las características de la ontología que pretende reutilizar. Estos problemas suelen obstaculizar la construcción de aplicaciones basadas en ontologías, aunque es previsible que, con el diseño de nuevos estándares de lenguajes *web*, se incremente notablemente la reutilización de ontologías en la llamada "*web* semántica" [Ber01].

Siguiendo la clasificación anterior de Uschold y Jasper [Usc99], se exponen aquí algunos

ejemplos bien conocidos de aplicaciones basadas en ontologías, algunas identificados por van Zyl y Corbett en [Zyl00a] y [Zyl00b] siguiendo este mismo escenario:

- En el escenario de realización neutral: destacan, por ejemplo, las aplicaciones que usan o extienden la ontología lingüística *GUM*, como *Penman* [Bat90], *KOMET* [Bat94], *TechDoc* [Ros94], *AlFresco* [Sto93] y *OntoGeneration* [Agu98]. El proyecto *KACTUS* [21] [Sch95] trata de modelar el conocimiento sobre diferentes sistemas técnicos creando ontologías de dominio y reutilizándolas para diferentes aplicaciones.
- En el escenario de acceso común a información: se puede mencionar el proyecto *Workflow Management Coalition* (WfMC) [Coa99] que proporciona una ontología, a modo de glosario de términos, que sirve como un vocabulario común sobre términos técnicos para vendedores de sistemas de gestión. Esta ontología está descrita de manera altamente informal en lenguaje natural, y sólo es utilizable por humanos. En el ámbito empresarial, *Enterprise Design Workbench* y *Integrated Supply Chain Management Project* [22] usan la ontología *TOVE*, y *Enterprise Project* [23] [Usc98b] usa la ontología *Enterprise*. En *KRAFT project* [Pre99], la ontología lingüística *Wordnet* es enlazada con varias ontologías compartidas para unificar la interpretación de términos lingüísticos. En el dominio del diseño de sistemas de información, *Comet* y *Cosmos* [Duk95]. El proyecto *Plinius* [24] [vVe94] utiliza una ontología para representar propiedades de materiales cerámicos para extraer conocimientos de manera semiautomática de textos escritos en lenguaje natural.
- En el escenario de indexación: podemos destacar el proyecto *IBROW3* [Ben98] que ha desarrollado un agente para la configuración y ejecución de sistemas de conocimientos según las preferencias de los usuarios. Los sistemas de conocimientos se configuran a partir de métodos de resolución de problemas que residen en librerías digitales en *Internet*. *Ontobroker* [25] [Fen98] y el proyecto *SHOE* [26] [Hef99] usan ontologías para anotar información en páginas HTML y mejorar la recuperación de información. En [Sta01] se presenta la forma de construir portales de conocimientos que usan ontologías, las cuales proporcionan los medios para el acceso y mantenimiento de tales conocimientos.

[21] http://www.swi.psy.uva.nl/projects/Kactus/Reports.html
[22] http://www.eil.utoronto.ca/
[23] http://www.aiai.ed.ac.uk/project/enterprise/
[24] http://wwwis.cs.utwente.nl:8080/kbs/pubpage.html
[25] http://www.aifb.uni-karlsruhe.de/WBS/broker
[26] http://www.cs.umd.edu/projects/plus/SHOE/spec.html

2.4. MARCOS DE TRABAJO PARA CLASIFICAR ONTOLOGÍAS

En esta sección se presentan características que algunos autores han elaborado sobre diferentes puntos de vista para comparar ontologías. Como se verá, estos enfoques de comparación y clasificación carecen de la completitud necesaria para que un usuario pueda escoger una frente a otras. Se profundiza en este aspecto porque este trabajo presenta una propuesta de características para describir ontologías, que posteriormente se utilizarán para compararlas.

El orden cronológico en el que aparecieron trabajos sobre características de ontologías es el siguiente: Gruber [Gru93a], en 1993, presentó el primer conjunto de características. Años más tarde, Uschold y Grüninger [Usc96b] presentaron en 1996 otros criterios que sirven como base al diseño y uso de ontologías. En 1997, Noy y Hafner [Noy97] presentaron un marco de 28 características para comparar ontologías. Simultáneamente, Hovy [Hov97] presentó su propuesta de 36 características en un documento de trabajo que no ha sido publicado. Por este motivo, algunas características propuestas no están claramente definidas. Posteriormente, en 1998, Uschold [Usc98a] presentó un marco de 10 características para clasificar aplicaciones basadas en ontologías que incluía algunas características de ontologías.

Los trabajos que se exponen en esta sección se han realizado, principalmente, con el objetivo de proporcionar a potenciales usuarios de ontologías información útil para ser empleada en el desarrollo de nuevos sistemas. Algunos trabajos, como el Gruber, y el de Uschold y Grüninger, identifican características fundamentales que deben tenerse en cuenta en el desarrollo de ontologías. Otros estudios, como los de Noy y Hafner, y el de Hovy, proponen un esquema común para comparar ontologías. Finalmente, en el trabajo de Uschold se identifican características de ontologías desde el punto de vista del papel que desempeñan en una aplicación.

Existen artículos en los que se muestran características para diseñar, describir, comparar y evaluar ontologías. De los artículos citados, los que muestran características de ontologías son: [Noy97], [Hov97] y [Usc98a]; y desde los que se pueden extraer características, aunque no tratan con profundidad estos temas, son [Gru93a] y [Usc96b].

2.4.1. Características Básicas de Diseño de Gruber

Uno de los primeros artículos que presentan criterios para construir ontologías es [Gru93a]. En dicho trabajo se proponen 5 criterios de diseño para el desarrollo de ontologías:

- **Claridad**: una ontología debería comunicar el significado de sus términos de una forma objetiva e independiente del contexto social o computacional.
- **Coherencia**: deberían analizarse si las inferencias son consistentes con las definiciones.
- **Extensibilidad**: debería ser diseñada anticipando los usos que se darán a su vocabulario.

Además, debería ser construida para que la extensión y la especialización se lleven a cabo de forma homogénea.

- **Mínima dependencia con respecto a la codificación**: debería permitir que los agentes que compartan los conocimientos puedan ser implementados en diferentes sistemas y estilos de representación.
- **Mínimo compromiso ontológico**: una ontología debería imponer las menores exigencias posibles sobre el dominio que modela, permitiendo a las partes comprometidas con la ontología tener libertad para especializar e instanciar la ontología como la necesiten.

2.4.2. Características Básicas de Diseño de Uschold y Grüninger

En el artículo de Uschold y Grüninger [Usc96b], que sirve como introducción al diseño y uso de ontologías, se identifican algunas características generales de las ontologías, como son:

- **Formalidad**: que expresa el nivel de formalidad por el cual se crea un vocabulario y se especifica su significado (*altamente informal* si está en lenguaje natural, *semi-informal* si está expresado en un lenguaje natural estructurado y restringido, *semi-formal* definido en un lenguaje formal artificial y *rigurosamente formal* si sólo está especificado para ser leído por los ordenadores).
- **Propósito**: que caracteriza los posibles usos de la ontología. El propósito lo subdivide en tres categorías: ***para comunicación*** (con el fin de reducir la confusión terminológica y conceptual al proporcionar un vocabulario consensuado y unificado. Dichas ontologías pueden tener los siguientes usos: establecer modelos normativos, crear redes de relaciones, mantener la consistencia y evitar la ambigüedad, e integrar varias perspectivas de diferentes usuarios); ***para interoperabilidad*** (usar la ontología como interlengua y para unificar herramientas *software*); y ***en Ingeniería de Sistemas*** (las ontologías mejoran la especificación, la fiabilidad y la reusabilidad).
- **Tema del dominio**: que describe la parte del mundo que quiere representar la ontología.

2.4.3. Esquema de Noy y Hafner

Noy y Hafner [Noy97] proponen un marco para comparar ontologías con el propósito de estudiar la diversidad de los diseños de ontologías e indicar las fortalezas y debilidades de cada uno. Las características las clasifican en los siguientes grupos: 1) generales, 2) relacionadas con el proceso de diseño, 3) sobre la taxonomía de los conceptos, 4) sobre la estructura conceptual interna de los conceptos, 5) sobre los axiomas, 6) sobre los mecanismos de inferencia, 7) sobre las aplicaciones que la usan, y 8) sobre las contribuciones de la ontología. El marco de

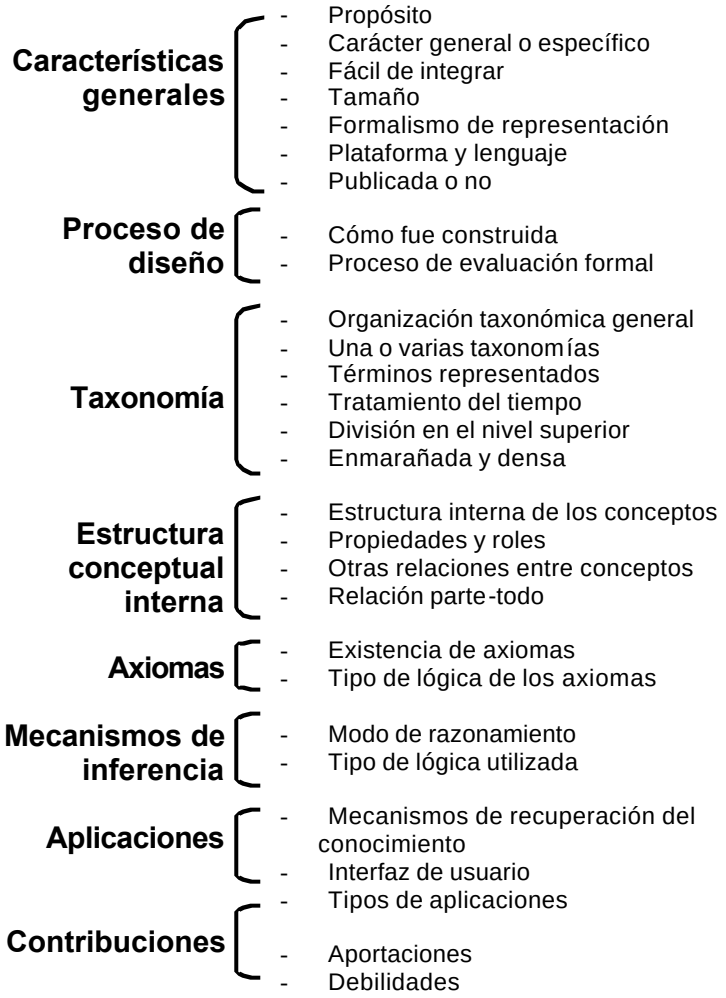


Figura 2.2. Taxonomía de características de Noy y Hafner [Noy97].

comparación, expuesto en la figura 2.2, fue aplicado a las siguientes ontologías: *CYC*, *Dahlgren's ontology*, *GUM*, *Gensim*, *KIF* [27], *Plinius*, *Sowa's Ontology*, *TOVE*, *UMLS* y *Wordnet.* El esquema de comparación tiene un total de 28 características organizadas de la siguiente forma:

1. **Características generales**: a) el **propósito** para el que la ontología fue creada; b) si es una ontología de carácter general o de **dominio** específico; c) si es **fácil de integrar** en otra ontología de tipo más general; d) su **tamaño**, identificado por el número de conceptos, reglas, axiomas, enlaces, etc.; e) los **formalismos** usados (marcos, lógica de primer orden, grafos conceptuales, redes semánticas, etc.); f) **plataforma y lenguaje de implementación**; y g) si está **publicada** y disponible.

2. Sobre el **proceso de diseño**: a) se estudia **cómo fue construida** la ontología (la

[27] A pesar de que en el estudio se incluye *KIF*, se debe señalar que *KIF* no puede considerarse como una ontología, sino como un lenguaje de implementación de ontologías.

aproximación de construcción, fuentes de conocimientos, etc.); y b) proceso de **evaluación formal** seguido.

3. Sobre la **taxonomía** de los términos: a) se examina la **organización taxonómica general** de la ontología; b) si hay **varias** taxonomías; c) sobre los **términos** representados: qué conceptos incorpora la ontología (cosas reales o abstractas, procesos y eventos, relaciones, y propiedades); d) el **tratamiento del tiempo**, es decir, si tienen conceptos temporales o no; e) la división del **nivel superior**: cómo divide el mundo en el nivel superior (abstracto vs. real, individual vs. colección, etc.); y f) sobre cómo de **enmarañada** o **densa** es la taxonomía (un único árbol de estructura, jerarquía múltiple, número de subcategorías en niveles superiores, etc.).
4. Sobre la **estructura conceptual interna y las relaciones entre conceptos**: a) si los conceptos tienen **estructura interna** (si tiene un conjunto de propiedades que le dan significado a cada categoría); b) si hay **propiedades y roles**; c) si hay **otras clases de relaciones entre conceptos** (espacial, funcional, etc.); y d) cómo se representan las relaciones **parte-todo** (como componente-objeto, miembro-colección, materia-objeto, característica-actividad, lugar-área, fase-proceso, etc.).
5. **Axiomas**: a) si hay **axiomas** para representar conocimientos diferentes a los taxonómicos, y b) **cómo son expresados** (qué tipo de lógica se utiliza para expresarlos, como lógica de primer orden, de segundo orden, lógica modal, etc.).
6. **Mecanismos de inferencia**: a) **modo** de razonamiento; y b) si utiliza **lógica de primer orden** u otro tipo.
7. Sobre las **aplicaciones** que usan la ontología: a) los **mecanismos de recuperación** del conocimiento que emplean; b) cómo es la **interfaz de usuario**; y c) el **tipo de aplicación** donde fue usada la ontología (procesamiento de lenguaje natural, recuperación de información, simulación y modelado, etc.).
8. **Contribuciones** de la ontología: a) las mayores **aportaciones** de la ontología; y b) sus **debilidades.**

2.4.4. Esquema de Hovy

Hovy propone en su trabajo [Hov97] una taxonomía de características bien estructurada para comparar ontologías, que fue realizada para el área de procesamiento de lenguaje natural. El esquema, mostrado en la figura 2.3, fue desarrollado para comparar ontologías lingüísticas. Aunque su propuesta no está terminada y nunca ha sido publicada, se presenta y se discute por

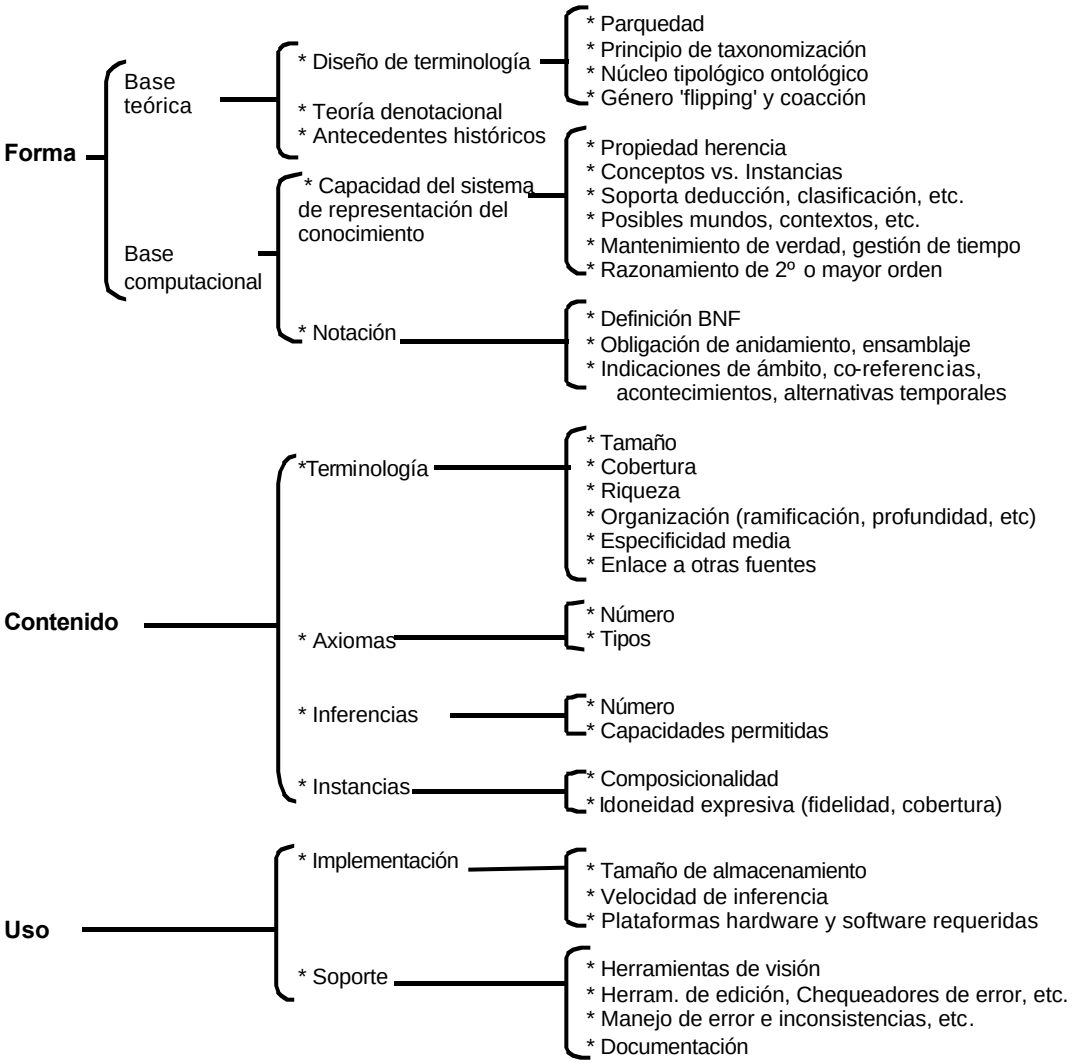


Figura 2.3. Esquema de características de ontologías de Hovy.

su extensión y novedad en el momento en el que se editó como informe técnico interno del instituto *ISI* [28]. Identifica 36 características de ontologías, aunque muchas de ellas no están descritas de forma precisa y algunas se mencionan para definir a otras. Los primeros niveles de su taxonomía describen lo siguiente:

- **Forma**: identifica el marco de representación y las herramientas conceptuales con las que el constructor de ontologías define los términos. Proporciona la base de representación e incluye aspectos teóricos, notacionales y computacionales.

  Con base teórica

  - **Principios del diseño terminológico**: Incluye todas las consideraciones que llevan a los constructores de ontologías a la selección de términos para modelizar aspectos

[28] http://www.isi.edu

del dominio.

- **La teoría denotacional** describe las teorías usadas para establecer el enlace formal entre la ontología y su extensión en el dominio. Las alternativas tienden a ser, desde informales, hasta extremadamente formales.
- **Los antecedentes históricos** incluyen todas las ontologías y sistemas de representación del conocimiento sobre los que la ontología se basa y desde los que son sacados los contenidos y las decisiones de diseño.

Con base computacional

- Las **propiedades y capacidades del sistema de representación del conocimiento** incluyen todos los aspectos relacionados con los soportes de representación y razonamiento requeridos por la ontología.
- La **notación** (formalismos) indica el diseño de notación usado para representar instancias en el dominio.

- **Contenido**: representa los términos usados por el constructor de ontologías para modelizar el dominio de interés.
  - La **terminología** incluye aspectos de los términos actuales incluidos por el constructor de la ontología, tales como: número de términos, organización, cobertura sobre el dominio, etc.
  - Los **axiomas** de la ontología, incluyendo cualquier otra interrelación definida entre los términos.
  - Las **reglas de inferencias** del dominio incluyen todos los aspectos de las reglas definidas sobre los términos en la ontología.
  - Las **instancias** indican aspectos de las instancias creadas para representar entidades del dominio en un tiempo y estado específicos.
- **Uso**: indica la manera con la que el constructor de ontologías se ocupa de la construcción, uso y mantenimiento de la ontología, incluyendo soporte software y herramientas para estas actividades.
  - **Implementación**: implica los aspectos relevantes de instanciaciones computacionales de la ontología.
  - **Soporte**: incluye aspectos del constructor de ontologías y de soporte a los usuarios.

2.4.5. Marco de Características de Uschold

Con el propósito de clasificar aplicaciones basadas en ontologías y así facilitar el uso de ontologías en aplicaciones similares, Uschold [Usc98a] propone un marco de 10 características. Algunas de ellas caracterizan a la ontología que soporta a la aplicación:

- **Propósito** para el que fue construida la ontología.
- **Lenguajes y paradigmas de representación** usados para representar la ontología.
- **Significado y formalidad**: cómo de extenso y formal es el significado de sus términos.
- **Tema del dominio**: si es de tipo general o de algún dominio específico.
- **Tamaño**: sobre cómo es de extensa la ontología, la aplicación o sus componentes.
- **Estado de desarrollo**: el grado en el que se encuentra la aplicación implementada.
- **Arquitectura conceptual**: cuáles son los principales componentes de la aplicación y cómo se relacionan entre ellos.
- **Mecanismos y técnicas** para usar la ontología: qué tipo de inferencia usa la aplicación, traductores, forma de acceder a los conocimientos, etc.
- **Plataforma de implementación**: qué lenguajes de implementación se han utilizado, herramientas *software*, sistema operativo necesario para usar la ontología, etc.
- **Miscelánea**: otros aspectos necesarios para usar la ontología con éxito.

2.4.6. Conclusiones sobre los Marcos de Trabajo Existentes para Clasificar Ontologías

En este apartado se han reunido diferentes estudios de características y clasificaciones de características identificadas con diversos propósitos. Debido precisamente a que el propósito por el que fueron consideradas es diferente, estos esquemas de características son difícilmente comparables. No obstante, las propuestas más elaboradas, y también más recientes, tienden a organizar taxonómicamente los conjuntos de características. Un resumen de las propuestas, con el número de características y el propósito para el que fueron creadas, se muestra en la tabla 2.1.

Autor	Año	Nº de características	Propósito
Gruber	93	5	Criterios de diseño
Uschold y Grüninger	96	3	Criterios de diseño
Noy y Hafner	97	28	Estudio de las diversas formas de diseño de ontologías
Hovy	97	36	Comparar ontologías lingüísticas
Uschold	98	10	Establecer el papel de las ontologías en las aplicaciones

Tabla 2.1. Resumen de los trabajos relacionados con características de ontologías.

Por un lado, las 5 características de Gruber [Gru93a], y las 3 características señaladas por Uschold y Grüninger [Usc96b], son de carácter muy general y fueron descritas como propiedades fundamentales que deben considerarse en el diseño de ontologías y que deben ser tenidas en cuenta cuando se plantee su reutilización.

El esquema de comparación de características de Noy y Hafner [Noy97] reúne 28

características sobre ontologías, aunque la definición poco precisa de algunas de ellas puede englobar a varias más. Con este marco se indicaron las diferencias y similitudes entre 10 ontologías escogidas. Las pretensiones del estudio eran comparar las diferentes alternativas de forma y diseño de ontologías para clarificar la descripción de un estándar de construcción de ontologías. Los valores que pueden tomar algunas características pueden ser bastante confusos. Además, en el marco identificado no se señalan los criterios para clasificar una ontología conforme a las características indicadas.

El marco de características de Hovy [Hov97] propone una taxonomía de 36 características sobre diferentes aspectos de las ontologías. Este marco fue planteado para poder comparar ontologías y reutilizarlas en sistemas de lenguaje natural. El estudio se realizó como documento interno de trabajo en el ISI, por lo que algunas características están poco o nada definidas, y a veces algunos conceptos carecen de sentido. Además, varias de las características identificadas sólo pueden ser rellenadas en ontologías lingüísticas.

Las 10 dimensiones de características indicadas por Uschold [Usc98a] fueron reunidas desde el punto de vista del papel que desempeñan las ontologías en una aplicación. El artículo presenta un apunte sobre características de aplicaciones y de las ontologías que proporcionan los conocimientos. La pretensión de creación de este marco era que nuevos constructores de aplicaciones basadas en ontologías pudieran utilizar esta información para desarrollar aplicaciones que tuvieran los mismos requisitos. Las características identificadas aquí carecen de precisión en su definición, y con el marco propuesto es difícil clasificar convenientemente las ontologías.

2.5. CARACTERÍSTICAS IDENTIFICADAS EN LAS DIMENSIONES PRINCIPALES DE REUTILIZACIÓN DE ONTOLOGÍAS

Además de los marcos de trabajo expuestos en la sección anterior, que diferentes autores han elaborado para clasificar y comparar ontologías, existen otros trabajos que también proporcionan características utilizadas para comparar lenguajes de implementación, metodologías de desarrollo de ontologías, y también entornos software de construcción de ontologías. Se incluyen tales propuestas en el estado de la cuestión de este trabajo porque a la hora de seleccionar una ontología para ser utilizada en una aplicación concreta, además del contenido que la ontología proporciona, el usuario debe analizar aspectos metodológicos y tecnológicos que serán cruciales a la hora de su integración en la aplicación. A continuación se presentan características en las siguientes dimensiones: contenido de la ontología, lenguaje de implementación de la ontología, metodología empleada en su construcción y entornos software

existentes para el desarrollo y edición de ontologías.

2.5.1. Características para Determinar el Contenido de las Ontologías

Un aspecto fundamental que un usuario debe examinar de una ontología para decidir reutilizarla es su universo de discurso. Gruber [Gru95a] define el universo de discurso como el conjunto de objetos que pueden ser representados cuando los conocimientos de un dominio se expresan mediante un formalismo declarativo. De esta forma, cuando un usuario encuentra una ontología, lo primero que debe hacer es analizar los términos que tiene representados y comprobar si coinciden con las necesidades del sistema que quiere construir. A esta dimensión Hovy [Hov97] la definió como **contenido**, aunque las características que determinó para describirlo sólo se centraban en algunos aspectos estructurales.

Se debe hacer notar que **no se ha encontrado en la literatura ningún esquema de características de ontologías planteado, expresamente, para que el usuario estime el grado con el que una ontología satisface el dominio requerido para su aplicación**.

A continuación se exponen algunos trabajos que pueden tener relación con características sobre el contenido de ontologías aunque, como ya se ha mencionado, ninguno tiene como fin la valoración de la adecuación del contenido de una ontología a un nuevo proyecto. En primer lugar se indican algunos artículos que identifican características para determinar la estructura de una ontología, y en el segundo apartado se apuntan otros que examinan características para identificar los términos existentes en las ontologías. En este apartado no aparecen los esquemas de Gruber [Gru93a] ni de Uschold y Grüninger [Usc96b], debido a que ambos son estudios que indican características generales sobre aspectos de diseño, y no aparece ninguna característica que pueda relacionarse con el contenido de las ontologías.

2.5.1.1. Características sobre Aspectos Estructurales del Contenido

Algunas de las publicaciones referenciadas en el apartado 2.4 incorporan características sobre aspectos estructurales en sus esquemas de clasificación de ontologías. Estas características se refieren al número de términos que especifican las ontologías, a características de las relaciones existentes, y sobre cómo está estructurada la taxonomía de conceptos. A continuación se indican las características identificadas en estos esquemas.

En el esquema de Noy y Hafner [Noy97]***:***
Identifican varias características relacionadas con la estructura, comparando las diferencias entre las diez ontologías.

- **Sobre el tamaño**: número de conceptos, relaciones, axiomas y reglas.
- **Sobre la taxonomía**: si está o no definida explícitamente, si es simple o compleja (respecto a la herencia), y cómo es de enredada y densa.
- **Sobre la estructura interna y relaciones entre conceptos**: qué clases de propiedades (atributos) están definidas, y qué tipo de relaciones existen. Hacen mención especial a las relaciones *parte-todo*.
- **Sobre axiomas:** si están definidos explícitamente, y si usan representaciones más complejas que la lógica de primer orden.

En el esquema de Hovy [Hov97]***:***
En el aspecto que denomina "contenido" enumera algunas características estructurales, muchas de ellas sólo aplicables a ontologías lingüísticas.

- **Sobre el tamaño**: número de términos.
- **Cobertura**: porcentaje de los conceptos del dominio representados por los términos.
- **Riqueza**: medida, por ejemplo, por el número de predicados por término, número de interconexiones entre términos, etc.
- **Organización**: relativo a la taxonomía y tipos de relaciones definidas.
- **Especificidad**: relativo al grado de abstracción de los términos.
- **Axiomas**: número de axiomas y tipos.
- **Reglas de inferencia**: número definido.
- **Instancias**: aspectos sobre las instancias existentes en la ontología.

En el esquema de Uschold [Usc98a]***:***
Las únicas características relacionadas con la estructura de contenido que menciona el esquema de Uschold son:

- **Escala:** relativa al tamaño de la ontología: número de conceptos y axiomas.
- **Arquitectura conceptual:** sobre las relaciones que existen entre los términos de la ontología.

2.5.1.2. Características Usadas para Identificar Términos de Ontologías

Existen trabajos que examinan características de las definiciones de los términos en las ontologías con el objetivo de mezclarlas. Weinstein y Birmingan [Wei99], y Maedche y Staab [Mae01] proponen métricas de coincidencia semiótica entre los componentes de dos ontologías. Estas métricas comparan características relacionadas con los nombres de los conceptos, de los atributos, de los valores que toman los atributos y las coincidencias en las taxonomías. Hovy

[Hov98] propone un conjunto de heurísticas para hallar la coincidencia de nombres de los conceptos y sus definiciones en lenguaje natural, y heurísticas para hallar la distancia semántica entre dos jerarquías de conceptos.

En el plano tecnológico, el entorno Chimaera [29] [MGu00] ayuda al usuario a unir múltiples ontologías importadas desde diferentes lenguajes. Sugiere potenciales términos candidatos de unificación basándose en el análisis de un número de características: el grado de equivalencia de los nombres de los conceptos, sus definiciones en lenguaje natural, las relaciones existentes entre términos y mediante la evaluación de la taxonomía de los conceptos.

2.5.1.3. Conclusiones sobre Características Relacionadas con el Contenido

En las publicaciones anteriores se han indicado características relacionadas con aspectos estructurales y para identificar términos de ontologías. Sobre aspectos estructurales del contenido de las ontologías se han encontrado algunas características en [Noy97], [Hov97], y [Usc98a] que hacen referencia, principalmente, al número de términos de la ontología, con qué relaciones se forma la taxonomía de conceptos y sobre características referentes a la disposición de la taxonomía de conceptos.

Para realizar emparejamientos de los términos existentes en varias ontologías, en Weinstein y Birmingan [Wei99], Maedche y Staab [Mae01], y Hovy [Hov98] se ofrecen métricas para valorar la coincidencia semántica de los términos y taxonomías de varias ontologías. El entorno Chimaera [MGu00] ofrece, además, una herramienta para realizar esta valoración de forma semiautomática. Los trabajos y la herramienta anterior exponen características y fórmulas interesantes para realizar procesos de mezcla del contenido de varias ontologías, pero no tratan la adecuación de los términos de las ontologías a un nuevo proyecto.

Así, se debe señalar que **no se han encontrado trabajos que propongan características de ontologías para analizar la coincidencia de los términos y su estructura, según las necesidades de un sistema**.

2.5.2. Características que Describen los Lenguajes de Implementación de Ontologías

Para poder implementar los conocimientos de un dominio en ontologías se han diseñado diferentes lenguajes, cada uno con unas capacidades expresivas y de razonamiento propios. Cuando un usuario analiza una ontología para incorporarla a su sistema, uno de los aspectos

[29] http://www.ksl.stanford.edu/software/chimaera/

fundamentales que debe estudiar es el lenguaje en el que se encuentra implementada. Esto es especialmente importante al integrar la ontología en el sistema. En algunos proyectos pueden requerirse lenguajes con gran capacidad expresiva, y en otros, puede necesitarse realizar inferencias con los conocimientos almacenados.

Por un lado, existen lenguajes "clásicos" de implementación de ontologías como *Ontolingua* [Far96], *CycL* [Len90], *LOOM* [MGr91], *FLogic* [Kif95], *OCML* [Mot95], *OKBC* [Cha97], etc. Por otro lado, en los últimos años, se están diseñando lenguajes de ontologías que se basan en los lenguajes de marcado de la *web*. Ejemplos de estos lenguajes son *XOL* [Kar99], *OML/CKML* [Ken98], *SHOE* [Luk00], *RDF(S)* [Las99], *OIL* [Hor00], y *DAML+OIL* [Hor01]. Van Harmelen y Fensel [vHa99] analizan aspectos sintácticos y semánticos de lenguajes de marcado *HTML*, *XML* y *RDF*. Además, en [MEn99] se evalúan diferentes lenguajes para utilizarlos en el campo de la bioinformática.

Considerando los dos grupos de lenguajes, Corcho y Gómez-Pérez [Cor00] han hecho un estudio, definiendo un amplio marco de características para evaluarlos, que se expondrá en este apartado. Este marco de trabajo, que sigue las dos dimensiones propuestas por la metodología *CommonKADS* [Sch99], clasifica las características en dos grandes bloques: expresividad del modelo de conocimientos del lenguaje y técnicas que caracterizan su mecanismo de inferencias.

- **Representación del conocimiento**: describe los elementos o componentes que se pueden utilizar en el lenguaje para representar los conocimientos de un dominio.
  - **Conceptos** o clases: son colecciones de objetos del dominio. Las características identificadas sobre conceptos son:
    - ¬ Posibilidad de definir metaclases: posibilidad de hacer clases como instancias de otras clases.
    - ¬ Posibilidad de definir clases disjuntas como particiones: clases hermanas que no pueden tener ninguna instancia común.
    - ¬ Mecanismos para definir atributos (*slots*). Pueden ser definidos como:
      - de instancia: atributos definidos en una clase que pueden tomar valores diferentes para cada instancia.
      - de clase: son atributos propios de la clase.
      - locales: atributos que afectan a un único concepto.
      - globales: atributos que se declaran una vez en la ontología y que, directamente, no se vinculan a ningún concepto.
      - polimórficos: atributos globales que pueden tener distinto comportamiento según el concepto al que se refieran.

¬ Facetas predefinidas de los atributos:

- valor por defecto: es el valor asignado si, al crear una instancia, no se ha indicado ninguno.
- tipo de valor: para obligar a asignar un tipo de dato determinado.
- cardinalidad: sobre el número mínimo y máximo de valores que puede tomar el atributo.
- documentación: que defina en lenguaje natural el atributo.
- definiciones operacionales: se podrán asociar fórmulas o reglas, que se activarán al realizar alguna acción (asignar, modificar o borrar) en el valor del atributo.
- posibilidad de añadir nuevas facetas a las predefinidas en el lenguaje.

o **Taxonomías**. Las taxonomías son usadas para organizar los conceptos del dominio usando relaciones de generalización y especialización. Las características identificadas son:

¬ Si está definida en el lenguaje la relación *subclase-de*, para especializar conceptos generales en otros más específicos.

¬ Permite descomposición disjunta de clases: un conjunto X de subclases de la clase Y es disjunta si una instancia sólo puede serlo de una de las clases de X; puede existir una instancia que lo sea de Y y no lo sea de ninguna de las clases de X.

¬ Permite descomposición exhaustiva de clases: un conjunto X de subclases de la clase Y es una partición exhaustiva si es una partición disjunta y no puede existir ninguna instancia de Y que no lo sea de una clase X.

¬ Está definida en el lenguaje la relación *no-subclase-de* para declarar que una clase determinada no es especialización de otra clase.

o **Relaciones**: representan interacciones entre conceptos del dominio. Suelen aparecer las relaciones del tipo: *subclase-de*, *parte-de*, *conectado-a*, etc. Pueden ser binarias o n-arias dependiendo de los argumentos que tomen. Las características identificadas son:

¬ Permite definir funciones.

¬ Permite definir relaciones/funciones n-arias: indica si es posible usar cualquier número de argumentos en las relaciones/funciones.

¬ Restringe el tipo de los argumentos que aparecen en las relaciones y funciones.

¬ Posee restricciones de integridad para comprobar la corrección del valor que toman los argumentos.

¬ Si existen definiciones operacionales asociadas a las relaciones y funciones para inferir valores de los argumentos por medio de reglas, fórmulas o procedimientos.

o **Axiomas**: son fórmulas, en ocasiones expresadas en lógicas de primer orden, que se utilizan para modelar sentencias que son siempre verdad y para verificar la corrección

de la información contenida en la ontología. Los axiomas también son usados para realizar inferencias sobre los conocimientos almacenados.

- ¬ Usa lógica de primer orden o segundo orden.
- ¬ Se pueden definir axiomas independientes de los conceptos: axiomas identificados con un nombre que no se asocian a ningún concepto, sino como un elemento independiente en la ontología.

o **Instancias/Hechos/Afirmaciones**: son términos usados para representar elementos concretos del dominio.

- ¬ Permite definir instancias [Gru93b]: representan elementos de un concepto determinado.
- ¬ Permite aseverar hechos: los hechos [Mot99] se suelen utilizar para representar una relación que se establece entre elementos.
- ¬ Maneja afirmaciones: las afirmaciones [Luk00] se refieren a la declaración de un hecho por parte de una instancia.

o **Reglas de Producción**: [MGr91] son reglas que siguen la estructura *Si «condición» entonces «acción»*, y se utilizan para expresar heurísticas de condiciones y acciones. Las características que se definen son:

- ¬ Permite definir premisas conjuntivas y disyuntivas en las reglas.
- ¬ Permite definir declarativamente el mecanismo de encadenamiento que se utilizará con las reglas: encadenamiento hacia atrás, hacia delante o ambos.
- ¬ Permite definir valores de verdad o certeza en las reglas.
- ¬ Permite incluir procedimientos en el consecuente; por ejemplo, para cambiar valores en los atributos de los conceptos.
- ¬ Las reglas, cuando son activadas, actualizan la base de conocimientos añadiendo o eliminando hechos, instancias o afirmaciones.

- **Mecanismos de inferencia**: describe cómo las estructuras estáticas pueden ser usadas para llevar a cabo un proceso de razonamiento. La forma en la que se ha representado el conocimiento va a condicionar, en gran parte, los métodos de razonamiento que pueden ser empleados. Las características identificadas en esta dimensión son:
  - ¬ Si el lenguaje posee un motor de inferencias implementado correcto y completo. Correcto indica que las soluciones alcanzadas por el motor de inferencias son siempre ciertas. Completo se refiere a que siempre proporciona una solución, si ésta existe.
  - ¬ Si el motor de inferencias realiza clasificaciones automáticas. Esto permite también detectar inconsistencias en taxonomías existentes.
  - ¬ Si el motor de inferencias maneja excepciones en la herencia de los atributos. Esta cualidad permite redefinir atributos en las subclases para señalar características particulares de las subclases.

+: disponible -: no disponible *+/-: no disponible de forma directa* ***D**: desconocido*		Ontolingua	OKBC	OCML	LOOM	FLogic	XOL	SHOE	RDF(S)	OIL	DAML+OIL
REPRESENTACIÓN DEL CONOCIMIENTO											
Conceptos	Definir metaclases	+	+	+	+	+	+	-	+	-	-
	Particiones	+	-	-	+	-	-	-	-	-	-
	Atributos de instancia	+	+	+	+	+	+	+	+	+	+
	Atributos de clase	+	+	+	+	+	+	-	+	+/-	+/-
	Atributos polimórficos	+	+	+	+	+	-	-	-	+	+
	Atributos locales	+	+	+	+	+	+	+	+	+	+
	Valores por defecto	+	+	+	+	+	+	-	-	-	-
	Tipo exigido en atributos	+	+	+	+	+	+	+	+	+	+
	Cardinalidad exigida en atributos	+	+	+	+	+/-	+	-	-	+	+
	Documentación	+	+	+	+	-	+	+	-	+	+
	Conocimiento operacional	-	-	+	+	-	-	-	-	-	-
	Añadir nuevas facetas	+	+	-	+	-	-	-	-	-	-
Taxonomías	Relación *Subclase_de*	+	+	+	+	+	+	+	+	+	+
	Descomposiciones exhaustivas	+	-	+/-	+	+/-	-	-	-	-	+
	Descomposiciones disjuntas	+	-	+/-	+	+/-	-	-	-	+	+
	Relación *No_Subclase_de*	+/-	-	-	+/-	-	-	-	-	+	+
Relaciones	Funciones	+	+	-	+	+	-	-	-	+	+
	Relaciones n-arias	+	+/-	+	+	+/-	-	+	+	+/-	+/-
	Tipos de conceptos restringidos	+	+	+	+	+	-	+	+	+	+
	Restricciones de integridad	+	+	+	+	+	-	-	-	-	-
	Definiciones operacionales	-	-	+	+	+	-	-	-	-	-
Axiomas	Lógica de primer orden	+	+/-	+	+	+	-	+/-	+/-	-	-
	Lógica de segundo orden	+	+/-	-	-	-	-	-	-	-	-
	Permite axiomas independientes	+	+	+	-	-	-	-	-	-	-
Instancias/ Hechos/ Afirmaciones	Definir Instancias	+	+	+	+	+	+	+	+	-	+
	Definir Hechos	+	+	+	+	+	+	+	+	+	+
	Definir Afirmaciones	+/-	-	-	-	-	-	+	+	-	-
Reglas de Producción	Premisas Conjuntivas	-	-	+	+	-	-	-	-	-	-
	Premisas Disyuntivas	-	-	+	+	-	-	-	-	-	-
	Valores de verdad en consecuente	-	-	-	-	-	-	-	-	-	-
	Ejecución de procedimientos	-	-	+/-	+	-	-	-	-	-	-
	Actualiza la base de conocim.	-	-	+	+	-	-	-	-	-	-
MECANISMOS DE INFERENCIA											
MI Completo		-	-	+	+	+	-	-	-	+	+
MI Correcto		+	-	-	-	+	-	+	+	+	+
Clasificaciones automáticas		-	-	-	+	-	-	+	-	+	+
Maneja excepciones		-	-	-	-	+	-	-	-	-	-
Herencia monótona		+	+	+	+	+	D	+	D	+	+
Herencia no monótona		-	+	-	+	+	D	-	D	-	-
Herencia simple		+	+	+	+	+	D	+	+	+	+
Herencia múltiple		+	+	+	+	+	D	+	+	+	+
Ejecuta procedimientos		+	+	+	+	-	-	-	-	-	-
Chequea restricciones		+	+	+	+	+	-	-	-	+	+
Encadenamiento hacia atrás y delante		-	-	+	+	+	-	D	-	-	-

Tabla 2.2. Resumen del estudio de Corcho y Gómez-Pérez [Cor00] sobre diez lenguajes de ontologías.

- Tipo de herencia permitida: simple, múltiple, monótona, no monótona.
- Si los procedimientos que se definen en el lenguaje son ejecutables o, por el contrario, son sólo definidos de forma declarativa.
- Los axiomas se utilizan para realizar comprobaciones de restricciones, con el objeto de mantener la consistencia de la ontología.
- Tipo de encadenamiento de reglas usado por el motor de inferencia: hacia delante, hacia atrás, o mixto.

Corcho y Gómez-Pérez, utilizando el marco de características anterior, analizan también en [Cor00] las capacidades expresivas y de razonamiento de nueve lenguajes, implementando una ontología de ejemplo en cada uno de ellos, con el fin de escoger uno o varios para representar un conjunto de ontologías de comercio electrónico en el proyecto *MKBEEM* (IST nº 1999-10589). En [Cor02], este marco se utiliza, además, para el análisis de lenguajes utilizados en la *web* semántica. La tabla 2.2 muestra un resumen de este análisis con valores actualizados sobre las características de los lenguajes, y se han incorporado los valores del lenguaje de implementación de ontologías *DAML+OIL* [Hor01].

2.5.3. Características que Describen las Metodologías de Desarrollo de Ontologías

Como en cualquier otro campo técnico, para construir ontologías de una manera fiable es muy aconsejable emplear una adecuada metodología, siguiendo los procesos de desarrollo, y las actividades y técnicas indicadas. En el campo de la Ingeniería del Conocimiento existen metodologías para construir SS.BB.CC., como la metodología de Waterman [Wat86], *CommonKADS* [Wie92] o *IDEAL* [Gom97]; y, en los últimos años, se han diseñado metodologías específicas en el campo de la Ingeniería Ontológica. Tal es el caso de la metodología de *CYC* [Len90], de Uschold y King [Usc95], Grüninger y Fox [Gru95b], la propuesta de Bernaras y colegas [Ber96], la metodología de *SENSUS* [Swa97], *METHONTOLOGY* [Fer97] y [Gom98], *Co4 Methodology* [Euz95], y *(KA)^2 Methodology* [Ben99]. Estas metodologías tienen las siguientes particularidades:

La metodología empleada en construir la ontología *CYC*, concebida ésta como una enorme base de conocimientos con conocimientos de sentido común, propone principalmente dos tareas para el desarrollo de ontologías: 1) desarrollo de una ontología de representación de conocimientos que contenga las primitivas de representación que se van a utilizar y 2) representación del resto de los conocimientos utilizando estas primitivas. La metodología no propone una fase de conceptualización ni de diseño, sino que los conocimientos se codifiquen

directamente con un lenguaje propio denominado *CycL*.

La metodología de Uschold y King indica unas líneas maestras que deben seguirse para desarrollar ontologías: 1) identificación del propósito para el que va a ser creada, 2) construcción de la ontología, siguiendo los pasos de: captura del conocimiento, codificación e integración de ontologías existentes, 3) evaluación de los productos obtenidos, y 4) documentación que debe llevar asociada el desarrollo de la ontología. Cabe destacar que no propone una fase de conceptualización del conocimiento ni de diseño formal.

La metodología de Grüninger y Fox propone construir un modelo lógico de conocimientos a partir de unos requerimientos. Básicamente, propone realizar una descripción informal de las especificaciones de la ontología para transformarla en una descripción formal. Las fases propuestas son: 1) determinar los escenarios que motivan la construcción de la ontología, 2) formulación de las consultas de suficiencia de manera informal, 3) especificación de la terminología que va a seguirse de manera formal, 4) formulación de las consultas de suficiencia de manera formal, 5) especificación de los axiomas y definiciones de términos de manera formal, y 6) definición de las condiciones bajo las cuales las soluciones de las consultas de suficiencia son completas.

La propuesta de Bernaras y colegas está concebida para construir ontologías mediante la abstracción e integración de bases de conocimientos concretas. El enfoque de esta metodología está condicionado por el desarrollo de aplicaciones, de forma que, cada vez que se construye una aplicación, se construye la ontología que representa los conocimientos necesarios para la aplicación. Si se decide reutilizar conocimientos de otras aplicaciones, las ontologías de dichas aplicaciones se modifican para hacerlas compatibles entre sí, y se incorporan a la ontología de la nueva aplicación.

La metodología *METHONTOLOGY* posibilita la construcción de ontologías en el nivel de conocimientos, y de manera independiente de las aplicaciones en las que se vayan a utilizar. Esta metodología proporciona: un proceso de desarrollo de ontologías adaptado del estándar IEEE 1074-1995 [IEE96], un ciclo de vida basado en prototipos evolutivos, y técnicas para llevar a cabo las actividades propuestas en el ciclo de vida. El proceso de desarrollo incluye las actividades que deben efectuarse cuando se construyen las ontologías: 1) actividades de gestión: control y garantía de calidad; 2) actividades orientadas al desarrollo: especificación, conceptualización, formalización, implementación y mantenimiento de la ontología; y 3) actividades de soporte: adquisición de conocimientos, integración, evaluación, documentación y administración de la configuración. *METHONTOLOGY* ha sido la metodología propuesta por la *Foundation for Intelligent Physical Agents* [30] (FIPA), la cual promueve la interoperabilidad entre las aplicaciones basadas en agentes.

[30] http://www.fipa.org

La metodología basada en *SENSUS* es usada para construir ontologías tomando como base los conceptos abstractos identificados en la ontología *SENSUS*. Mediante esta metodología, la construcción de una ontología en un determinado dominio se lleva a cabo en dos etapas: 1) poda de los términos no relevantes de la ontología *SENSUS*; para ello, se parte de unos términos relevantes del dominio que se toman como semilla y se excluyen todos los que no se encuentran en el camino hasta la raíz; y 2) se incluyen en este grafo otros términos relevantes del dominio.

Co4 [Euz95] es un protocolo para buscar consenso entre varias bases de conocimientos. Su propósito es que la gente pueda discutir y comprometerse sobre los conocimientos introducidos en la base de conocimientos de un sistema. Estas bases de conocimientos son construidas para ser compartidas, y tienen conocimientos consensuados, por lo que pueden considerarse como ontologías. Las bases de conocimientos se organizan en un árbol, en el que en las hojas se incluyen los usuarios de las bases de conocimientos, y los nodos intermedios son los grupos de bases de conocimientos. La idea es que, cuando un usuario tiene suficiente confianza en parte del conocimiento de su base de conocimientos, busca el consenso sobre estos conocimientos con el resto de los usuarios de su grupo.

La iniciativa $(KA)^2$ propone crear ontologías sobre la comunidad de adquisición de conocimientos, desarrolladas en un esfuerzo conjunto por un grupo de gente de diferentes localizaciones, usando las mismas plantillas y lenguaje. La ontología $(KA)^2$ formará la base para anotar documentos *web* de la comunidad de adquisición de conocimientos y proporcionar acceso inteligente a esos documentos. $(KA)^2$ fue una iniciativa abierta donde los participantes se involucran activamente en el desarrollo de la Ingeniería Ontológica de forma distribuida.

En [Fer99b], Fernández hace un estudio de las siguientes metodologías de desarrollo de ontologías: la metodología de Uschold y King, la metodología Grüninger y Fox, la propuesta de Bernaras y colegas, *METHONTOLOGY*, y la metodología de *SENSUS*, estableciendo una serie de criterios como marco de comparación, que son los siguientes:

- **Herencia de la Ingeniería del Conocimiento**: influencia de los enfoques de Ingeniería del Conocimiento en la metodología de desarrollo de ontologías.
- **Detalle de la metodología**: consideración del grado de detalle en el que las actividades y técnicas propuestas por la metodología están especificadas.
- **Recomendaciones para la formalización del conocimiento**: formalismos propuestos para representar los conocimientos del dominio.
- **Estrategia para construir ontologías**: se refiere a la relación existente entre la ontología y la aplicación que la va a explotar. Las clasifica en: dependiente, semidependiente e independiente de la aplicación.

	Uschold & King	Grüninger & Fox	Bernaras et al.	METHONTOLOGY	Basado en SENSUS
Herencia de metodologías de IC	Parcial	Poca	Grande	Grande	Ninguna
Detalle de actividades y técnicas	Muy poco	Nada	Muy poco	Muy detallado la mayor parte	Medio
Recomendaciones para la formalización del conocimiento	Ninguno	Lógica	Ninguno	Ninguno	Redes semánticas
Estrategia para construir ontologías	Independiente de la aplicación	Semidepend. de la aplicación	Dependiente de la aplicación	Independiente de la aplicación	Semidepend. de la aplicación
Estrategia para identificar conceptos	Media	Media	De arriba a abajo	Media	De abajo a arriba
Ciclo de vida recomendado	Ninguno	No definido totalmente	Ninguno	Prototipos evolutivos	No definido totalmente
Diferencias con IEEE 1074-1995	Faltan procesos y actividades	Faltan procesos y actividades	Faltan procesos y actividades	Faltan procesos predesarrollo y actividades	Faltan procesos y actividades
Técnicas recomendadas	Desconocido	Desconocido	Desconocido	Faltan algunas técnicas	Desconocido
Ontologías desarrolladas	Enterprise Ontology	TOVE	Ontología para redes eléctricas	CHEMICALS | Ontologías para medio ambiente | Reference Ontology | $(KA)^2$ reestructurada	SENSUS

Tabla 2.3. Resumen del estudio de Fernández [Fer99b] sobre metodologías de desarrollo de ontologías.

		Uschold & King	Grüninger & Fox	Bernaras et al.	METHONTOLOGY	Basado en SENSUS
Procesos de dirección del proyecto		No declarados	No declarados	No declarados	Las actividades del entorno del proyecto no están identificadas	No declarados
Procesos orientados al desarrollo del proyecto	**Procesos pre-desarrollo**	No declarados	No declarados	No declarados	No declarados	No declarados
	Procesos de desarrollo	Declarados los procesos de requerimientos e implementación, pero no los de diseño	Declarados	Declarados	Declarados	Declarados los procesos de requerimientos e implementación, pero no los de diseño
	Procesos post-desarrollo	No declarados	No declarados	No declarados	No identifica actividades para instalación, operación, soporte, desinstalación y formación	No declarados
Procesos integrales		No identifica las actividades para formación, estudio del entorno, y dirección de configuración	No identifica las actividades para formación, estudio del entorno, y dirección de configuración	No declarados	No identifica las actividades para formación y estudio del entorno	No declarados

Tabla 2.4. Resumen del estudio de Fernández [Fer99b] sobre la conformidad de las metodologías de desarrollo de ontologías con el estándar IEEE 1074-1995.

- **Estrategia para identificar conceptos**: ésta puede ser [Usc96b]: desde los más concretos a los más abstractos, desde los más abstractos a los más concretos, o una aproximación intermedia que combina ambas, también llamada desde dentro hacia fuera.
- **Ciclo de vida recomendado** por la metodología.
- **Diferencias con el estándar IEEE 1074-1995**: se comprueba si los procesos y actividades del estándar son soportados por la metodología.
- **Técnicas recomendadas**: si se proponen técnicas particulares para cada actividad recomendada en el ciclo de vida.
- **Ontologías desarrolladas con la metodología y sistemas que utilizan dichas ontologías**.

Con estos criterios se compararon las metodologías de desarrollo de ontologías anteriormente citadas, cuyo resumen puede verse en la tabla 2.3. En la tabla 2.4 se presentan las metodologías analizadas desde la perspectiva del IEEE 1074-1995.

2.5.4. Características que Describen los Entornos de Desarrollo de Ontologías

Al igual que existen lenguajes para representar ontologías y metodologías para desarrollarlas, en la actualidad existen aplicaciones software específicas para el desarrollo de ontologías. Tales herramientas suelen tener módulos para importar y exportar las ontologías especificadas desde/en diferentes lenguajes. Además, algunas de ellas dan soporte a algunas de las actividades de las metodologías indicadas en el apartado anterior. Entre los entornos software más conocidos y utilizados se encuentran: el entorno de desarrollo de la ontología *CYC* [Len90], *Ontolingua Server* [Far96], *Ontosaurus* [Swa97], *Tadzebao-WebOnto* [Dom98], *ODE* [Fer99a] y *WebODE* [Arp01], *OntoEdit* [Sta00], *OILed* [Bec01], y *Protégé2000* [Noy01].

Duineveld y colegas [Dui99] proponen un marco de evaluación para caracterizar 6 entornos de desarrollo de ontologías: *Ontolingua Server, Tadzebao-WebOnto, ProtégéWin* (versión previa de *Protégé2000*), *Ontosaurus, ODE* y *KADS22* [31], cuyo resultado se resume en la tabla 2.5. En la tabla no incluyeron *KADS22* por ser un entorno en fase de desarrollo. En este artículo, además, presentan un lugar de consulta en *Internet* denominado *Wondertools* [32] para examinar propiedades de entornos de desarrollo de ontologías.

[31] http://www.swi.psy.uva.nl/projects/kads22/
[32] http://www.swi.psy.uva.nl/wondertools

+: disponible -: no disponible +/-: difícil de usar NA: no aplicable	Ontolingua	WebOnto	ProtégéWin	OntoSaurus	ODE
Generales					
Claridad de la interfaz	-	+	+	-	-
Consistencia de la interfaz	+	+	+	+	+
Velocidad de actualización	-	+/-	+	-	+
Visión general de la ontología	+/-	+	+	+	-
Claridad de los comandos	+	+	+	+	+/-
Identificación de cambios	+/-	+/-	+/-	+/-	+/-
Estabilidad	+	+	+	+	-
Instalación local	No	No	Sí	No	Sí
Sistema de ayuda	+	-	+	+	-
Sobre el desarrollo de la ontología					
Permite herencia múltiple	Sí	Sí	Sí	Sí	Sí
Permite descomposición exhaustiva y disjunta	+	+	-	+	+
Verificación de la consistencia	+	+	+	+	+
Ejemplos disponibles	+	+	+/-	+	+/-
Librerías de ontologías disponibles	+	+	-	+	-
Existencia de primitivas de alto nivel	+	+	-	+	+
Información sobre los términos usados	-	-	-	-	+
Sobre el trabajo en cooperación					
Edición síncrona por varios usuarios	+	+	-	+	-
Bloqueo de la ontología	+	+	-	+	-
Permite visualizar con la ontología bloqueada	+	+	NA	+	NA
Cambios de los usuarios son reconocibles	-	-	-	-	-
Permite exportar a varios formatos	+	-	-	+	+/-
Permite importar desde otra herramienta	+	-	-	+	+

Tabla 2.5. Estudio de Duineveld y colegas [Dui99] sobre cinco entornos de desarrollo de ontologías.

El marco de evaluación de [Dui99] plantea las siguientes características:

- **Características generales**: a) claridad de la interfaz; b) consistencia de la interfaz; c) velocidad de actualización; d) visión general de la ontología; e) claridad de los comandos; f) identificación de cambios; g) estabilidad; h) instalación local; e i) sistema de ayuda.
- **Características vinculadas con el desarrollo de la ontología**: a) permite herencia múltiple; b) permite descomposición exhaustiva y disjunta; c) verificación de la consistencia; d) ejemplos disponibles; e) librerías de ontologías disponibles; f) existencia de primitivas de alto nivel; y g) información sobre los términos usados.
- **Características sobre el trabajo en cooperación**: a) edición síncrona por varios usuarios; b) bloqueo de la ontología; c) permite visualizar con la ontología bloqueada; d) los cambios de los usuarios son reconocibles; e) es posible exportar a varios formatos; y f) es posible importar desde otras herramientas.

En febrero de 2001, en el marco de la red temática *OntoWeb*, se celebró un *workshop* [33] con el objetivo de definir las características que debería tener un editor de ontologías universal. Entre las propuestas, cabe destacar las características esbozadas en [Bro01] referentes a este entorno:

- **Escalabilidad**: relativa a la facilidad para adquirir a gran escala instancias y relaciones, que múltiples usuarios puedan leer y escribir, que potencie la modularización, etc.
- **Adquisición de instancias y axiomas**: de forma que compruebe los tipos e inconsistencias de instancias y axiomas, tender a la adquisición automática de instancias, desarrollar un creador de axiomas de forma gráfica.
- **Visualización**: buscar la claridad de visualización de los diferentes términos, sobre todo en la taxonomía, proporcionar información sobre el contexto cuando sea necesario, etc.
- **Modularización**: controlar las versiones de las ontologías, promover la mezcla e integración, e inclusión en entornos visuales.
- **Transparencia del razonamiento**: los motores de inferencia deben ser ajustables, deben tener un depurador de errores, y deben tener *tests* de eficiencia.
- **Multitarea y características similares a las bases de datos**: el conocimiento debe estar *on-line*, disponible para ser consultado por varios agentes, deben mantener el bloqueo en los procesos de lectura/escritura, debe tener índices de búsqueda para mejorar la rapidez, y otras características usuales en los sistemas de gestión de bases de datos.
- **Competencia**: la herramienta deberá realizar *tests* de validación y verificación sobre los términos incluidos en la ontología.

El *workshop* sirvió para fijar los primeros pasos de construcción del editor universal. Aunque no se especificaron de forma concreta, algunas de las características que se buscan del editor es que tenga el código abierto, con un entorno parecido a *Protégé2000* [Noy01], y capaz de trabajar con *DAML+OIL*, *RDF*, y otros lenguajes *web* de implementación.

2.5.5. Conclusiones sobre las Clasificaciones de Características de Ontologías

En los anteriores apartados se han recopilado artículos relacionados con características de contenido, lenguajes, metodologías y entornos software para desarrollar ontologías. Como ya se indicó en la sección 2.4.6, los trabajos de Gruber [Gru93a], Uschold y Grüninger [Usc96b], Noy y Hafner [Noy97], Hovy [Hov97] y Uschold [Usc98a] ayudan a clasificar ontologías sólo parcialmente; pero no son útiles para comparar la idoneidad de una o varias ontologías candidatas con respecto a las necesidades de una aplicación, debido a que estos esquemas no

[33] *OntoWeb Workshop*: *"Universal Ontology Editor Workshop"*, Vrije Universiteit, Amsterdam (Países Bajos), 13 de febrero de 2001. (http://www.ontoweb.org/download/deliverables/feb13-2001.pdf)

fueron concebidos para este fin. Por ello, **ninguno de los marcos encontrados sirve para decidir cuál de las ontologías encontradas es la mejor para usarla en un sistema**.

En el apartado 2.5.1 se identifican publicaciones relacionadas, aunque parcialmente, con los aspectos semánticos y estructurales del contenido de las ontologías. Los artículos referenciados sólo tratan aspectos para realizar la correspondencia semántica entre términos de varias ontologías, y otros identifican algunas características relacionadas con la estructura taxonómica de los términos de las ontologías; pero **ninguno sirve para que el usuario estime el grado con el que el contenido de una ontología satisface las necesidades de contenido de su sistema**.

El lenguaje en el que se haya implementado la ontología es una de las dimensiones más importantes para que el usuario decida seleccionarla. El proyecto en el que se va a integrar la ontología puede estar implementado en un lenguaje que no es compatible o no cubre las necesidades del lenguaje en el que está implementada la ontología. El marco propuesto por Corcho y Gómez-Pérez [Cor00] permite realizar adecuadas comparaciones entre las capacidades de representación y razonamiento de los lenguajes de ontologías; pero **no se establece ningún método que permita, con la taxonomía de características propuesta, analizar el grado en el que los diferentes lenguajes satisfacen las especificaciones del proyecto**.

La metodología seguida para desarrollar y mantener la ontología es otra de las dimensiones que el usuario debe examinar para decidir usar una ontología. Una ontología creada siguiendo una adecuada metodología será más fiable que otra construida sin ninguna metodología. Además, el proceso para incorporar nuevos términos será más fácil si la metodología tiene bien especificadas las actividades y técnicas que deben realizarse. El estudio más sistemático, en cuanto a características de metodologías se refiere, es el realizado por Fernández en [Fer99b]. En él se propone un marco de características para comparar metodologías de desarrollo de ontologías. Sin embargo, **no se propone ningún método que indique lo conveniente que es una ontología desarrollada con una metodología frente a otra ontología desarrollada con otra, o más aún, sin ninguna metodología**.

Existen en la actualidad varios entornos software para desarrollar ontologías. La calidad de las herramientas vinculadas a la ontología será otra de las dimensiones que hará que el usuario decida seleccionar la ontología o no. El usuario puede requerir modificar términos, trabajar de forma conjunta con otros usuarios, integrar otras ontologías, mezclar varias ontologías, implementar la ontología en otro lenguaje, u otras acciones que se podrán realizar dependiendo de las características de las herramientas software. A pesar de que en [Dui99] y Brown [Bro01] se indican algunas características sobre entornos de desarrollo de ontologías, **estos marcos sólo apuntan algunos aspectos para valorar entornos de desarrollo de ontologías, y no proponen ningún método para analizar su idoneidad frente a las necesidades de uso en un nuevo proyecto**.

2.6. MÉTRICAS DE REUTILIZACIÓN

En este apartado se hace un análisis de las métricas existentes en Ingeniería del Software e Ingeniería del Conocimiento relacionadas con la reutilización de componentes y conocimientos. Se recopilan, en la primera sección, los modelos relacionados con la reutilización de componentes software, y en la segunda, los relacionados con la reutilización de conocimientos, y en particular con la reutilización de ontologías.

Como en cualquier campo de ingeniería, los procesos de medición llevados a cabo sobre proyectos software son fundamentales para hacer estudios o valoraciones sobre diversos aspectos del desarrollo de estos sistemas. Las mediciones del software pueden servir para comprender y controlar lo que sucede en la producción y para mejorar las características del proceso y del producto. En el campo de la economía del software, la industria reclama estimaciones de coste y esfuerzo en el proceso de desarrollo de sus productos.

La medición es el proceso por el que números o símbolos son asignados a atributos de entidades en el mundo real, de forma que lo describen de acuerdo a reglas claramente definidas [Fen96]. En la Ingeniería del Software, la idea de medir atributos del producto y del proceso ayuda a hacerlos más comprensibles y controlables. La medición en el software puede ser hecha usando atributos internos, considerados como cualidades puras del producto, y mediante atributos externos, que son las cualidades del producto relacionándolo con el entorno. Como atributos internos se pueden considerar las líneas de código (LOC) de un módulo, puntos de función, número de páginas de documentación, número medio de llamadas entre módulos, etc., y como atributos externos se pueden considerar la facilidad de manejo del usuario, el número de errores encontrados en un periodo de tiempo, la velocidad de ejecución, etc. A menudo se miden los atributos internos para predecir los atributos externos, ya que algunas cualidades intrínsecas de un producto pueden dar indicios de su comportamiento futuro en un entorno determinado.

2.6.1. Métricas en la Ingeniería del Software

Existen en la actualidad multitud de modelos y métricas que examinan y predicen atributos del software en todo su ciclo de vida [34]. La finalidad de las métricas es mejorar las prestaciones del producto y optimizar los procesos de producción en todas sus fases. La proliferación de estos modelos y métricas demuestra su importancia en el campo de la Ingeniería del Software. Se

[34] Por ejemplo en http://irb.cs.tu-berlin.de/~zuse/metrics/3-hist.html y http://www-ivs.cs.uni-magdeburg.de/sw-eng/us/bibliography/bib_main.shtml se reúnen varias clasificaciones, congresos y tendencias sobre métricas de software.

mencionan a continuación algunos de los modelos más usados y consolidados encontrados en la literatura, siguiendo la clasificación realizada por Fenton [Fen96]:

- Estimación de coste y esfuerzo: intentan predecir los costes del proyecto en fases tempranas del ciclo de vida; como el modelo *COCOMO* [Boe81] o *COCOMO II* [Abt98], el modelo *SLIM* de Putnam [Put78], o el modelo de puntos función de Albrecht [Alb79].
- Modelos y medidas de la productividad de la plantilla de desarrollo: como en los modelos *COCOMO* o los estudios de Jones [Jon86]
- Modelos y medidas de calidad: miden la calidad del software desde diversos puntos de vista. Se pueden incluir los factores de calidad de McCall [MCa77], los modelos *COCOMO*, el modelo *COQUAMO* de Kitchenman y Walker [Kit89], la evaluación *SPR* [Jon85] o el modelo ISO 9126 [Fen95]. Estudios detallados de varios modelos de calidad bien conocidos se encuentran en [Kan87].
- Modelos de fiabilidad: intentan predecir cuándo fallará un sistema; como el modelo de Jelinski-Moranda [Jel72], el modelo de Musa [Mus75] o el modelo de Littlewood [Lit73].
- Evaluación y modelos de ejecución: miden el tiempo de respuesta y el de producción de resultados (podrían ser incluidos en los modelos de calidad, pero son un grupo importante de modelos de medición). Entre ellos están los trabajos de Kleinrock [Kle75], Ferrari [Fer83] o Harel [Har92].
- Métricas de estructura y complejidad. Midiendo estos atributos internos tratan de predecir otros externos como la fiabilidad o la facilidad de mantenimiento. Dos ejemplos serían los modelos de McCabe [MCa76] o de Halstead [Hal77].
- Valoración de capacidad-madurez del software, para evaluar la calidad de los requerimientos de un proyecto. Generalmente basados en el modelo de madurez de la capacidad del *SEI* [Dem89], como el *SPICE* [Rou95].
- Aunque no se incluye como una categoría en la clasificación de Fenton, deben mencionarse las métricas propias de orientación a objetos que se recogen en [Pia95] y [Rio96].

Es aconsejable realizar las apreciaciones de características relevantes del producto lo antes posible, para que los responsables del proyecto puedan redirigir el proceso de producción y puedan optar por las mejores alternativas [Pre97]. Debe tenerse en cuenta que la Ingeniería del Software conlleva procesos muy complejos, con muchas variables entrelazadas, por lo que los procesos de valoración deben ser "razonablemente" precisos; como dijo Boehm [Boe81]: "...no podemos estimar el coste de producir 100.000 LOC de forma tan precisa a la de producir 100.000 aspirinas o 100.000 botes de *ketchup*...". O, dicho de otra forma, los modelos de valoración de atributos del software deben realizarse debido a la importancia que tienen para la

industria del software; pero teniendo en cuenta que, a veces, el nivel de precisión puede ser menor del que fuera deseable.

2.6.2. Métricas de Reutilización en la Ingeniería del Software

Según un estudio realizado en 1996, el Departamento de Defensa de los Estados Unidos podría haber ahorrado unos 300 millones de dólares anuales incrementando su nivel del reutilización del software en sólo un 1% [Ant96]. A pesar de que cada vez la industria del software realiza la reutilización de manera más usual, aún no se ha llegado a emplearla de forma sistemática. La reutilización del software no puede hacerse de forma aislada y acometerla en un determinado instante de tiempo; es necesario un marco de aplicación de Ingeniería del Software efectiva y una política de empresa que apueste por la reutilización [Pre97]. Para obtener con éxito esta meta se aplican varios tipos de métricas que analizan tanto los posibles beneficios de aplicar la reutilización, como el estudio de las condiciones que pueden hacer a un componente reutilizable.

Varios de los modelos que se muestran en el apartado anterior pueden ser aplicados a los modelos de reutilización del software. Se ha escogido la clasificación que hace Frakes [Fra96a] de estos modelos y métricas de reutilización:

- *Modelos coste-beneficio de reutilización:* incluyen el análisis coste/beneficio económico, y de calidad y productividad al reutilizar. Como el modelo de Gaffnet y Durek [Gaf89], la mejora de calidad de Barnes-Bollinger [Bar91] o las métricas de reutilización en negocios de Poulin y colegas [Pou93].
- *Valoración de la madurez:* valoran los programas software según el grado en el que aplican la reutilización del software. Como los trabajos de Koltun-Hudson [Kol91], y el modelo de capacidad de Davis [Dav91].
- *Cantidad de reutilización:* son métricas para valorar y controlar las mejoras obtenidas por la reutilización mediante porcentajes de reutilización en el tiempo en el ciclo de vida de los objetos. Se pueden mencionar el nivel de reutilización de Frakes y Terry [Fra94a], la fracción de reutilización de Agresti-Evanco [Agr92] y las métricas de orientación a objetos de Bieman-Karunanithi [Bie93].
- *Análisis de los modos de fallo:* identifica y clasifica los impedimentos para reutilizar en una organización determinada. Como el método de Frakes y Fox [Fra96b].
- *Evaluación de la reusabilidad:* indica el grado en el que un artefacto puede ser reutilizable. Se incluyen la clasificación por facetas de Prieto-Díaz y Freeman [Pri87], la medición para la reusabilidad de componentes *Ada* de Basili y colegas [Bas90], la

valoración de reusabilidad de *REBOOT* [Kar95], y la predicción de reutilización para objetos en el ciclo de vida de Frakes y Fox [Fra95].

- *Métricas de librerías de reutilización:* son usadas para controlar y seguir el uso de un repositorio de reutilización. Como las métricas de costes de indexación y efectividad de búsqueda de Frakes y colegas [Fra90][Fra94b] y las métricas de calidad de *assets* de Frakes y Nejmeh [Fra87].

2.6.3. Métricas de Reutilización en la Ingeniería del Conocimiento

Los modelos expuestos en el apartado 2.6.2 sobre reutilización del software también pueden aplicarse a la reutilización de conocimientos ya que, aunque con características específicas, los conocimientos son un tipo más de componente software. En la Ingeniería del Conocimiento, el proceso de adquisición de conocimientos conlleva un enorme esfuerzo de tiempo y recursos, por lo que su reutilización nos ofrece una gran ventaja. A pesar de su gran importancia, no se han encontrado publicados métodos específicos que aborden este problema. El proceso de decisión de reutilizar, o no, conocimientos se suele dejar al criterio del ingeniero del conocimiento que, basándose en su experiencia, decide si es conveniente reutilizar o, por el contrario, adquirir, conceptuar, formalizar e implementar los conocimientos desde el principio.

La industria del conocimiento no se ha convertido, por ahora, en un mercado estándar de compra/venta de conocimiento "preenvasado" mediante el uso de ontologías, como esperaban Neches y colegas [Nec91]. La reutilización de conocimientos encontró una barrera importante en la falta de traductores que pudieran resolver el problema de la heterogeneidad de los lenguajes y formalismos de representación de los conocimientos. En períodos sucesivos se paliaron algunas de las barreras tecnológicas para poder llevar a cabo la reutilización de ontologías en nuevas aplicaciones (Uschold [Usc98a] hace mención de esta idea citando el *workshop* sobre ontologías de *IJCAI'95*). En la actualidad, el número de aplicaciones que utilizan ontologías es grande, pero el número de aplicaciones que reutilizan ontologías desarrolladas para otros sistemas no lo es tanto. **La inexistencia de métodos que lleven a un usuario a decidir si una ontología existente es adecuada para su aplicación le puede disuadir de plantearse el proceso de su reutilización**. Es decir, si un usuario no dispone de ningún método que le permita analizar la idoneidad de las ontologías existentes y le pueda valorar cómo se ajustan a sus necesidades, puede rechazar la opción de reutilizarlas y optar por desarrollar una nueva.

Una de las metodologías más completas que describen el ciclo de vida asociado a la construcción de SS.BB.CC. es la metodología *IDEAL* [Gom97]. La metodología *IDEAL* proporciona al ingeniero del conocimiento un marco bastante completo para desarrollar

SS.BB.CC. mediante un ciclo de vida en espiral cónico en tres dimensiones, donde cada fase finaliza con el desarrollo de un prototipo que, si está bien construido, conduce a la siguiente fase del ciclo de vida, pero permitiendo la incorporación sistemática de nuevos conocimientos en cada "vuelta" de recorrido de la espiral, refinando el conocimiento acumulado, fruto del uso del propio sistema. Además, describe explícitamente los procesos de evaluación que deben ser llevados a cabo en cada una de las fases. Aunque la metodología *IDEAL* es una de las propuestas para el desarrollo de SS.BB.CC. que más aspectos considera, no dispone de un mecanismo específico para la evaluación del proceso de reutilización del conocimiento a partir de otros SS.BB.CC.

La mayoría de las metodologías propias de construcción de ontologías (cuya comparación puede verse en [Fer99b]) consideran la reutilización de ontologías existentes como uno de sus procesos clave para ser realizados. Especialmente es indicado en las metodologías de Bernaras y colegas [Ber96] y la metodología basada en *SENSUS* [Swa97]. En cambio, todas las metodologías analizadas adolecen de una técnica apropiada para medir la adecuación de una ontología a un nuevo sistema.

En [Pin00], Pinto y Martins proponen una serie de guías que se deben seguir para realizar el proceso de integración de ontologías. La integración [Pin99] se refiere al proceso de ensamblar, extender, especializar y adaptar ontologías existentes. Una de las actividades que proponen en el proceso de integración es la evaluación de las ontologías candidatas. Sugiere una serie de criterios que expertos en el dominio deben tener en cuenta: conocimientos olvidados, conocimientos eliminados, conocimientos cambiados de lugar, fuentes de conocimientos modificadas, documentación modificada, terminología cambiada, definiciones cambiadas y procesos modificados. Esta serie de criterios se deja al juicio de los expertos del dominio, pero no proponen ningún modelo ni métrica que se deba aplicar para ponderar la idoneidad de las ontologías candidatas al proceso de reutilización.

2.6.4. Métodos de Decisión Multicriterio. El Método de las Jerarquías Analíticas

En este apartado se define qué es un método de decisión multicriterio y se describe en particular el método de las jerarquías analíticas (AHP como acrónimo de *Analytic Hierarchy Process* [Saa77]). La justificación de este apartado es que AHP se ha escogido como la base teórica en la que se ha fundamentado el método propuesto en este trabajo, que consiste en una adaptación de AHP para el proceso de elección de ontologías.

Los métodos de decisión multicriterio tratan de resolver una cuestión general, que puede ser

descompuesta en problemas de decisión más específicos y, una vez resueltos, se pueden utilizar las soluciones parciales para resolver el objetivo inicial [Fen96]. De una manera formal, se puede definir un conjunto de objetos o **candidatos** *A*, cuyos elementos podrán ser elegidos como la solución final. Para este conjunto *A* se definen **criterios** *c*, que serán funciones que relacionen el conjunto de candidatos *A* con otro totalmente ordenado *O*. Por ser *O* totalmente ordenado, existirá una relación *R* entre cada par de elementos de *O* que sea reflexiva, transitiva, antisimétrica y fuertemente completa [Fen96]. Estas condiciones garantizan que cualquier candidato puede ser comparado con otro usando un criterio sobre *O*. Si una familia de criterios definidos sobre un conjunto de candidatos *A* es consistente [Roy90], entonces el problema de decisión multicriterio permite [Fen96]:

- Determinar el candidato o un subconjunto de candidatos de *A* que se consideren "el mejor" respecto a una familia de criterios: *problema de elección*.
- Dividir *A* en subconjuntos, de acuerdo a algunas normas: *problema de clasificación*.
- Categorizar los candidatos de *A* desde el mejor al peor: *problema de categorización*.

Teniendo en cuenta con más detalle el **problema de elección**, en el método de decisión multicriterio se tendrá un vector de criterios $<c_1, c_2, ..., c_n>$ que servirá para obtener una valoración de la bondad de cada elemento (candidatos) de *A*. Para ese vector de criterios se busca obtener la mejor valoración o la más razonable, y obtener así la elección del "mejor candidato". Se puede potenciar el valor de un criterio con respecto a otro, asignando unos ***pesos de substitución de criterios*** de forma subjetiva, pero controlada con respecto a alguna preferencia de juicio. Las formas en las que un decisor puede asignar pesos de importancia a cada criterio son: a) asignando directamente un peso normalizado a cada uno de los criterios, o b) indicando la importancia que tiene cada criterio en relación con los demás, y obteniendo, a partir de estas comparaciones, los pesos de importancia. A este último caso pertenece el método de las jerarquías analíticas que se describe en el siguiente apartado.

El Método de las Jerarquías Analíticas

El *Analytic Hierarchy Process* o método de las jerarquías analíticas ([Saa77] y [Saa90]) es uno de los métodos de decisión multicriterio más populares. Ideado por Thomas L. Saaty [35] en la década de los 70, AHP constituye una manera poderosa y flexible para la toma de decisiones en problemas multidimensionales o multicriterio complejos. Este método ayuda a las personas en

[35] Saaty ha creado la empresa *Expert Choice Inc.* que explota comercialmente con éxito este método; se puede acceder a su servidor en la URL: http://www.expertchoice.com/

la captura de conocimientos sobre el problema en cuestión, cuantifica los juicios subjetivos y fuerza al evaluador a comparar alternativas con respecto a los criterios establecidos.

La representación que hace AHP del problema de decisión es mediante un **árbol jerárquico de criterios**, donde el nodo raíz es el objetivo que se intenta resolver, los nodos intermedios del árbol son los criterios y subcriterios de decisión, y como nodos hoja están los candidatos que sirven como alternativas. El método propone los siguientes pasos:

PASO 1: Establecer el problema y sobre qué objetivo se va a tomar la decisión. El objetivo se situará en el nodo raíz del árbol jerárquico de criterios. En el ejemplo que se muestra en la figura 2.4, el objetivo es "elegir el mejor coche".

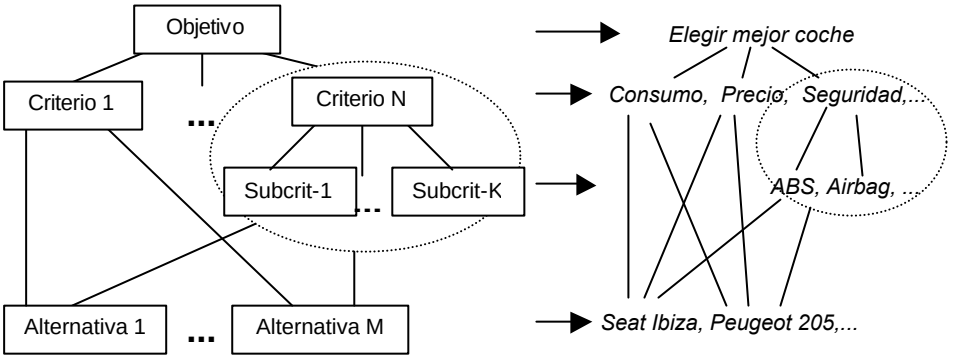


Figura 2.4. Estructura jerárquica del método AHP, y ejemplo del objetivo: "*Elegir el mejor coche*".

PASO 2: Organizar un árbol jerárquico de criterios (como aparece en la figura 2.4) del siguiente modo: El nodo raíz es el objetivo del problema, los niveles intermedios son los criterios que influyen en cada uno de los criterios padre del árbol, y en el nivel inferior se sitúan las posibles alternativas. Esta ordenación jerárquica sirve para apreciar los criterios y sus relaciones, inherentes al problema.

PASO 3: Para cada nodo padre (que tiene algún conjunto de nodos hijo) del árbol, se construye un conjunto de matrices de comparación entre los miembros hermanos (hijos del mismo nodo padre). Un evaluador hace una comparación de importancia por juicio de pares a los nodos de hermanos, denotando su contribución a cada uno de los nodos padre a los que está enlazado. La comparación entre dos criterios se hará indicando, con alguna escala numérica, la proporción de importancia de un criterio frente al otro. En la tabla 2.6 se muestra un ejemplo de valores relativos de importancia que un evaluador ha dado a los criterios del primer nivel identificados en la figura 2.4.

	Consumo	Precio	Seguridad	...
Consumo	1	1/6	2	
Precio	6	1	3	
Seguridad	1/2	1/3	1	
...				

Tabla 2.6. Ejemplo de matriz de comparación de criterios del primer nivel para "Elegir el mejor coche". El evaluador considera que, para él, el criterio "Consumo" es seis veces menos importante que el "Precio", dos veces más importante que la "Seguridad", etc.

Con estas matrices se busca obtener los pesos de importancia relativa de cada criterio frente a sus hermanos. Los pesos son obtenidos calculando los autovectores de las correspondientes matrices de comparación de pares. Para calcular los autovectores se resuelve el sistema de ecuaciones $|A - \lambda I| = 0$; siendo A la matriz de comparación, I la matriz identidad y λ los autovalores. Se elige el máximo autovalor. El autovector correspondiente a este autovalor resuelve la ecuación, obteniéndose los pesos de importancia de cada elemento, teniendo en cuenta que la suma de los pesos debe ser 1. Este proceso debe repetirse para cada nivel del árbol. La descripción del desarrollo matemático para el cálculo del autovector de pesos puede consultarse en [Rio89].

PASO 4: Para cada alternativa que vaya a ser considerada como posible solución, se asignan valores (con una escala prefijada) a cada uno de los nodos hoja del árbol (criterios situados en el nivel inferior). Siguiendo el ejemplo, la alternativa "*SEAT Ibiza*" deberá tener asignado un valor en el criterio "Seguridad en el *Airbag*" con una escala prefijada, otro valor en el criterio "Seguridad del *ABS*" con su escala correspondiente, etc. Todas las alternativas identificadas como candidatas deberán tener valoradas, con sus correspondientes escalas, los criterios situados en los nodos hoja.

PASO 5: Los valores asignados en el paso 4, junto a los vectores de pesos calculados en el paso 3, se combinan con algún tipo de fórmula de ponderación. Los resultados parciales se irán propagando por el árbol de manera ascendente hasta obtener el valor final del objetivo para cada una de las alternativas. Con estos valores de idoneidad, el usuario podrá tomar una decisión frente al objetivo fijado.

2.6.5. Conclusiones sobre las Métricas de Reutilización

En la Ingeniería de Software existen numerosas métricas que estudian diferentes aspectos sobre reutilización. Desde las predicciones coste/beneficios de reutilización, hasta las que miden la reusabilidad de un componente, estas métricas y modelos proponen características que pueden ser observadas para evaluar diferentes aspectos de la reutilización de componentes. En cambio, **no se han encontrado métodos concretos para valorar la idoneidad de un componente en**

una aplicación determinada, a pesar de los beneficios que se obtendrían de estas medidas.

En la Ingeniería del Conocimiento **no se han encontrado métricas específicas sobre la reutilización de conocimientos**. Sin embargo, sí se han encontrado algunos trabajos y metodologías que identifican características que deben ser observadas para reutilizar ontologías, aunque ninguno aborda el problema de cómo analizar características relevantes de ontologías para decidir su reutilización en un sistema concreto.

Por otro lado, los métodos de decisión multicriterio sirven para comparar varias alternativas cuando deben tenerse en cuenta, a la vez, varios objetivos. En estos métodos, el decisor puede asignar un peso normalizado directamente a un criterio, que indicará la importancia que tiene ese criterio para el objetivo final. Pero la normalización y composición de pesos con respecto a más de un criterio, medido sobre una misma escala estándar, lleva a obtener números sin sentido, porque normalizar conjuntos de números separados destruye la relación lineal que hay entre ellos [Fen96]. Para evitar esto, el método AHP propone, en primer lugar, comparar la importancia relativa que tiene cada criterio con respecto a todos los demás; a partir de estas valoraciones, calcular los pesos de los criterios, y finalmente normalizarlos para poder obtener las medidas para las alternativas existentes [Saa90]; por este motivo, **el método de las jerarquías analíticas constituye una de las mejores opciones para tomar decisiones multicriterio**.

2.7. CONCLUSIONES SOBRE EL ESTADO DE LA CUESTIÓN

En este capítulo se han presentado cuestiones relacionadas con ontologías y métricas de Ingeniería del Software e Ingeniería del Conocimiento. En la primera sección del capítulo se presentaron las ontologías más relevantes y las aplicaciones más conocidas basadas en ontologías, que han servido como fuente de conocimientos para este trabajo. A continuación, se hizo un estudio de los marcos de trabajo existentes y se realizó un análisis de las publicaciones que muestran características relacionadas con aspectos básicos que deben ser considerados en reutilización de ontologías, como son: el contenido de la ontología, los lenguajes en los que están implementadas, las metodologías y los entornos de desarrollo. Por último, se realizó un resumen de las métricas existentes en Ingeniería del Software e Ingeniería del Conocimiento que realizan valoraciones sobre la reutilización de componentes software y los basados en conocimientos. También se exponen los pasos de AHP, por ser el método en el que se basa este trabajo.

Del contenido de este capítulo pueden identificarse los siguientes problemas y carencias:

Problema 1. **Dispersión de las fuentes de información sobre ontologías**. Se emplea mucho tiempo en buscar ontologías disponibles para usarlas en una aplicación. La forma actual en la que un usuario busca información sobre ontologías existentes es examinando publicaciones científicas o, más comúnmente, accediendo a las URLs de: *Ontology.Org Library LinkBank*, *The Semantic Web*, *OntoWeb* y *DAML*; y consultando la información en los servidores *web* de ontologías, como: *Ontolingua Server, SHOE, ON9*, *DAML Ontology Library*, *WebODE, WebONTO* y *Ontosaurus*. Además de este consumo de tiempo, el usuario no tendrá la seguridad de que la búsqueda haya sido completa, es decir, que haya recopilado las ontologías sobre el dominio que está buscando, ya que la información puede estar obsoleta o pueden aparecer nuevos portales que el usuario desconozca.

Problema 2. **Diferente granularidad en la especificación de las ontologías**. La información suministrada en la *web* (y también en otras fuentes) describen las ontologías desde puntos de vista diversos, en diferentes lenguajes y con diferente grado de detalle. Al usuario le resulta complejo analizar esta información, ya que ésta puede encontrarse dentro de un texto extenso, dentro de la propia implementación de la ontología o puede no aparecer. Por ejemplo, preguntas como: ¿aparece el término *t* en la ontología?, ¿puedo utilizar la relación de tipo *R*?, con el lenguaje de implementación ¿puedo representar axiomas?, ¿permiten las herramientas de edición trabajar de forma cooperativa?, etc. Muchas preguntas de este tipo, que puede demandar el usuario, actualmente no pueden ser respondidas de forma simple, examinando únicamente las fuentes disponibles.

Problema 3. **Carencia de un marco unificado de trabajo de características de ontologías con el que poder hacer comparaciones**. Existen diversos trabajos que identifican características sobre diversos aspectos de ontologías, pero ninguno de ellos está desarrollado bajo el punto de vista del usuario que busca la ontología más apropiada para su nuevo sistema. No existe un esquema global que tenga en cuenta diferentes aspectos, relacionados con contenidos, lenguajes, metodologías y entornos software, los cuales deben ser tenidos en cuenta de forma conjunta para decidir reutilizar una ontología en una aplicación. Si se han conseguido reunir varias ontologías que tienen unas propiedades que se consideran indispensables, el usuario no dispone de un marco unificado de trabajo que le indique, de forma estructurada y completa, las características que debe examinar de las ontologías candidatas. Hacer una comparación sin seguir un esquema común de características puede llevar a que el usuario no considere todos los aspectos que necesite de la ontología, y que tampoco tenga en cuenta las características que influyen en estos aspectos.

Problema 4. **Carencia de un método para la toma de decisiones en la elección de ontologías**. Las métricas de reutilización existentes en Ingeniería del Software no sirven para valorar la idoneidad de las ontologías que pueden reutilizarse en aplicaciones, ya que, a pesar de que en Ingeniería del Software existen varias métricas que miden y estiman varios aspectos de reutilización de componentes, no existe un método que le indique al usuario lo adecuado que es un componente para incorporarlo a su proyecto. En Ingeniería del Conocimiento sólo se han encontrado guías o recomendaciones para llevar a cabo la reutilización de ontologías, pero no se ha encontrado ninguna métrica de reutilización. Actualmente, no existe ningún método que ayude al usuario a decidir la elección de unas ontologías frente a otras, por lo que esta decisión es totalmente artesanal y, por lo tanto, difícilmente justificable frente a directivos o clientes del proyecto.

Se puede concluir que, actualmente, el proceso de reutilización de ontologías está muy basado en la intuición. Los usuarios que buscan ontologías tienen muchas dificultades para encontrarlas y posteriormente compararlas, ya que la información disponible no suele encontrarse con el mismo grado de detalle, y no existe un marco global de características para poder compararlas. Los usuarios tampoco tienen un método que puedan seguir para evaluar si las posibles ontologías candidatas son apropiadas para su sistema. Así, la decisión de usar una u otra suele estar basada en su experiencia y, una vez tomada la decisión, es difícilmente justificable. Para resolver estos problemas, en el presente trabajo se propone **un método que permite a los usuarios determinar la idoneidad de ontologías existentes respecto a las necesidades de su proyecto.**

Capítulo 3

Planteamiento

3.1. INTRODUCCIÓN

En el estado de la cuestión se ha realizado un estudio de la situación actual de las ontologías, se han recopilado los marcos de clasificación de ontologías existentes, se han reunido las publicaciones que identifican características relacionadas con las dimensiones principales de uso de ontologías, se han expuesto las métricas que tratan aspectos de reutilización en Ingeniería del Software e Ingeniería del Conocimiento, y se han analizado las principales dificultades y carencias que un usuario se encuentra a la hora de seleccionar ontologías para incorporarlas a su sistema. En este capítulo se define el objetivo de este trabajo, se presentan las principales aportaciones y se indica la metodología de trabajo empleada. Finalmente, se incluyen las hipótesis de trabajo.

3.2. OBJETIVOS Y APORTACIONES DEL TRABAJO

El presente trabajo ofrece un método para obtener una valoración de idoneidad que ayudará al usuario [36] a evaluar, de forma justificada, cómo se adecuan una o varias ontologías a las necesidades de un nuevo sistema. Esta valoración de idoneidad se realizará ponderando, previamente, unos criterios de decisión según las directrices del nuevo proyecto y las preferencias del usuario. Las aportaciones que ofrece este método, esquematizadas en la figura 3.1, pueden resumirse de la siguiente forma:

A1. Se ha establecido un **marco multinivel de características de ontologías agrupadas por dimensiones.** Se pretende así:

- Reducir los tiempos de búsqueda de ontologías disponibles en la *web*.
- Establecer la granularidad de la descripción de ontologías existentes, a partir de un conjunto de propiedades.

[36] Los usuarios del método serán ingenieros del conocimiento especializados en el campo de las ontologías. Se mantiene el término "usuario" conviniendo que serán los responsables en la dirección del proyecto.

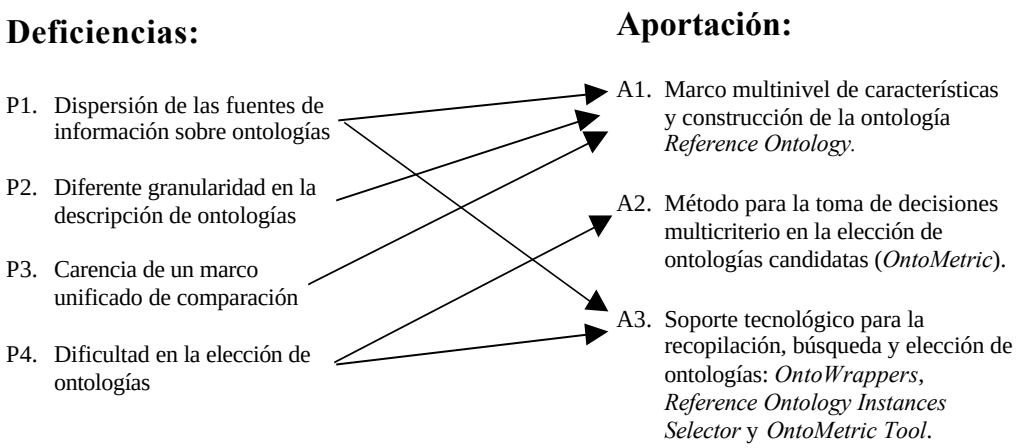


Figura 3.1. Relación entre las deficiencias encontradas con las aportaciones.

- Establecer un marco de características, agrupadas por dimensiones, que permite realizar comparaciones entre los criterios que deben ser analizados para elegir las ontologías candidatas.

Las dimensiones que se han identificado en el marco de características son: el contenido de la ontología, los lenguajes en los que se encuentra implementada, la metodología que se ha seguido para su construcción, los entornos software de desarrollo de ontologías, y los costes que ocasionará su uso. Cada una de estas dimensiones tendrá asociado un conjunto de criterios que servirá para describirlas, y que servirán al método *OntoMetric* para obtener las medidas de idoneidad. Estos criterios se han organizado en un marco multinivel de características, construido desde el punto de vista del usuario que se plantea usar una ontología en su sistema, y también servirá para analizar y comparar las ontologías candidatas. Es **multinivel** porque está estructurado de forma que el usuario podrá descender, en los criterios que considere necesarios, a niveles más profundos de detalle que describirán con más precisión esos criterios. Los valores de las características identificadas en este marco de características pueden obtenerse a partir de la información incluida en una ontología en el dominio de las ontologías denominada ***Reference Ontology*** (RO), que ha sido instanciada con los conocimientos de las respectivas dimensiones.

A2. Se ha desarrollado **un método que permite valorar la idoneidad de varias ontologías candidatas para incorporarlas a un sistema**. Así, el usuario puede decidir, de una forma justificada, qué ontologías entre varias candidatas son las más adecuadas. Con ello se pretende paliar el problema de la elección de ontologías para integrarlas en un proyecto.

El método propuesto, denominado ***OntoMetric***, basado en la toma de decisiones multicriterio, describe un conjunto de procesos para seleccionar ontologías candidatas, pondera la importancia de los criterios de decisión, y asigna una medida cuantitativa de

idoneidad a cada ontología candidata con relación al proyecto que la va a usar. El método utiliza el marco multinivel de características como esquema de criterios de decisión.

A3. **Se da soporte tecnológico a la recopilación y búsqueda de ontologías, y a los procesos de medición del método *OntoMetric***. Se han desarrollado herramientas software para solucionar principalmente tres problemas:

- Se han desarrollado un conjunto de aplicaciones que extraen la información de ontologías existentes en la *web* y en servidores de ontologías, y la transforma en instancias válidas del modelo conceptual de la RO. Estas aplicaciones se llaman ***OntoWrappers***.
- Para evitar el esfuerzo de la búsqueda de ontologías candidatas, debido a la dispersión de las fuentes de información, se ha creado la aplicación ***Reference Ontology Instances Selector*** (ROIS), que utiliza la ontología RO. Esta aplicación permite que el usuario que busca ontologías formule consultas compuestas (sobre varias características en varias dimensiones) para indicar los valores obligatorios en sus propiedades, que deben contener las ontologías. ROIS proporciona, tras estas consultas, el conjunto de ontologías candidatas que satisfacen las especificaciones indicadas. Un ejemplo de consulta sobre la aplicación ROIS podría ser: muéstrame las ontologías que tengan algún concepto con nombre o en su descripción en lenguaje natural, la palabra "mueble", que tengan un concepto con nombre "silla" en alguno de los 3 primeros niveles de la taxonomía, que tengan un número de relaciones mayor que 20, que estén representadas en un lenguaje que sea de marcado *web* y que permita definir axiomas, que puedan ser importadas por entornos software que permitan el trabajo cooperativo, que no cuesten más de 30.000 $, y que hayan sido desarrolladas siguiendo *METHONTOLOGY*.
- Para asistir al usuario en los procesos de medición del método *OntoMetric* y para decidir cuáles de las ontologías candidatas son las más adecuadas, se ha creado la aplicación ***OntoMetric Tool*** (OT). Esta aplicación utiliza las ontologías candidatas proporcionadas por ROIS, y ayuda al usuario a elegir, de entre ellas, las ontologías más adecuadas, mediante la adaptación del método AHP.

3.3. METODOLOGÍA DE TRABAJO

Se indican en este apartado las tareas que permiten alcanzar los objetivos propuestos:

Para obtener el **marco multinivel de características y construir la RO** se proponen las

siguientes tareas:

Tarea 1. Conceptualización del marco multinivel de características de ontologías. Esta tarea toma, como punto de partida, los trabajos sobre características que describen el contenido de las ontologías, lenguajes de implementación, metodologías y entornos de desarrollos de ontologías, y costes de uso, expuestos en las secciones 2.4 y 2.5. A estas características se han añadido otras para completar el marco. El resultado de esta tarea será un marco de comparación de ontologías que sirve como base de trabajo para el método propuesto.

Tarea 2. Validación del marco multinivel de características. Las características identificadas en la tarea anterior, organizadas en varias dimensiones y niveles en cada dimensión, se revisan mediante un cuestionario de opinión. En él, diversos usuarios experimentados en el uso de ontologías valoran el grado de importancia de cada característica en cada dimensión, e indican los defectos y omisiones encontrados.

Tarea 3. Formalización de una ontología en el dominio de las ontologías. Para poder obtener los valores del marco multinivel de características, se ha formalizado la ontología ***Reference Ontology***, que fue creada usando la metodología de desarrollo de ontologías *METHONTOLOGY* [Fer99a] y [Gom98], y utilizando como entorno software de desarrollo *WebODE* [Arp01].

Tarea 4. Instanciación de la RO. Este proceso se ha realizado de dos formas:

4.1. Los valores de las características sobre la dimensión de "contenido" de ontologías se han instanciado analizando las ontologías existentes y disponibles en servidores de ontologías: *Ontolingua Server*, desde el servidor *WebODE*, desde las páginas web de *SHOE* y del *Ontology Group at ITBM-CNR*, y desde el repositorio *DAML Ontology Library*. Esta instanciación se realiza de forma automática mediante los *OntoWrappers*.

4.2. Las instancias de las dimensiones de lenguajes, metodologías, entornos de desarrollo y costes de uso de ontologías se han completado a partir de publicaciones sobre estas dimensiones y examinando directamente los lenguajes, metodologías y entornos software.

Para construir el **método de decisión multicriterio *OntoMetric***, concebido para valorar las ontologías candidatas según las directrices del proyecto, se proponen las siguientes tareas:

Tarea 5. Selección de ontologías. Se han identificado los pasos para que un usuario pueda, a partir de las características representadas en la RO, seleccionar las ontologías que tengan unos valores obligatorios en estas características, y que serán consideradas como candidatas para ser elegidas.

Tarea 6. Adaptación del método AHP para la elección de ontologías. Se ha escogido el método AHP usado para la toma de decisiones multicriterio, y se ha adaptado al uso de ontologías, incorporando escalas de valoración lingüísticas. Estas escalas se corresponden con intervalos difusos que permite al método *OntoMetric* realizar los cálculos para obtener medidas de idoneidad cuantitativas sobre las ontologías analizadas.

Tarea 7. Identificación de los pesos por defecto en los criterios de decisión. Se ha querido ofrecer al usuario una ponderación inicial para cada una de las características del marco. Para obtener estos valores, se han utilizado los resultados del cuestionario mencionado en la tarea 2. A partir de estos resultados, se han obtenido los pesos medios de importancia a cada criterio, que servirán a los usuarios como referencia inicial, y que deberán ser ajustados según las necesidades de su proyecto concreto.

Para desarrollar la **plataforma tecnológica** que asiste en la aplicación del método *OntoMetric*, se han realizado las siguientes tareas:

Tarea 8. Implementación de las aplicaciones software para instanciar la dimensión "contenido" de ontologías, usando ontologías disponibles en diferentes lenguajes de implementación. Se han creado diferentes aplicaciones software, denominadas ***OntoWrappers***, que extraen la información relevante desde los ficheros donde se encuentran implementadas las ontologías, en sus correspondientes lenguajes, y las transforman en instancias válidas de la dimensión "contenido" de la RO. Los lenguajes seleccionados han sido: *Ontolingua*, *DAML* y *X-WebODE*, por ser estos los lenguajes en los que están implementadas las ontologías encontradas en los repositorios con ontologías disponibles: *Ontolingua Server* (*Ontolingua*), en el servidor *WebODE* (*X-WebODE*), en las página *web* de *SHOE* (*DAML*), en la del *Ontology Group at ITBM-CNR* (*Ontolingua*) y en la de *DAML Ontology Library* (*DAML*).

Tarea 9. Implementación de la aplicación software para la selección de ontologías. Se ha creado una aplicación *web* denominada ***Reference Ontology Instances Selector*** (ROIS) que permite hacer búsquedas en las instancias de la RO, y obtener las ontologías candidatas a ser utilizadas.

Tarea 10. Implementación de la aplicación software para la medición de idoneidad de ontologías. Se ha creado una aplicación software que asiste en el proceso de medición del método *OntoMetric*, llamada ***OntoMetric Tool*** (OT). Con OT se podrán modificar los criterios de decisión y sus pesos de importancia según las necesidades del sistema, y realizará los cálculos para obtener la medida de idoneidad para cada ontología candidata. Los valores de cada característica de cada ontología podrán ser completados a partir de la información proporcionada por ROIS. El usuario podrá usar los pesos de importancia por defecto, obtenidos a partir de los cuestionarios.

Tarea 11. Evaluación del método *OntoMetric* con casos de estudio. Utilizando ROIS y OT, se han realizado varios casos de estudio que han permitido evaluar la aplicabilidad del método. Por un lado, se han realizado casos de prueba individuales en cada una de las dimensiones por separado, y por otro, se han realizado casos de prueba considerando todas las dimensiones de forma conjunta.

La descripción de cómo se han llevado a cabo estas tareas se realiza en el capítulo 4 de este trabajo. En primer lugar se ofrece una visión general de los procesos del método *OntoMetric* junto con su solución tecnológica en los apartados 4.2 y 4.3. En el apartado 4.4 se describe el marco multinivel de características y el desarrollo de la ontología *Reference Ontology*. En el apartado 4.5 se describe el proceso de instanciación de esta ontología en todas sus dimensiones. El proceso de selección de ontologías candidatas se describe en el apartado 4.6, junto con la herramienta software construida para realizar esta tarea. Por último, el apartado 4.7 detalla el método de medición de idoneidad de las ontologías candidatas y la herramienta software que ayuda en este proceso. El método utiliza el marco multinivel de características, el modelo conceptual de la RO y sus instancias, y una adaptación del método de las jerarquías analíticas. En el capítulo 5 se muestran varios ejemplos de los casos de prueba realizados para evaluar la aplicabilidad del método, y en el capítulo 6 se indican las conclusiones del trabajo y líneas futuras.

3.4. HIPÓTESIS DE TRABAJO

Las **premisas** que se asumen a lo largo de esta memoria son las que se indican a continuación:

1. Los usuarios que van a utilizar el método deben ser personas que tengan conocimientos básicos en el uso de ontologías, aunque no es necesario que sean expertos en la materia. Los programas software que asisten en la aplicación del método poseen el esquema de características y sus pesos por defecto para ayudar a los usuarios menos experimentados.
2. La medida de idoneidad que se obtenga para cada una de las ontologías candidatas va a depender directamente de las metas especificadas del proyecto que se va a construir, por lo que antes de aplicar la métrica deberán estar perfectamente delimitados los objetivos y pretensiones del proyecto.
3. El marco multinivel de características propuesto no es rígido, sino que está pensado para ser dinámico y ajustable a las necesidades del proyecto. Es necesario tener identificados los objetivos del proyecto y la importancia de cada uno de ellos, antes de modificar el esquema de características para el proyecto en cuestión.

4. Cada vez que se cree un repositorio de ontologías (que se considere de interés) que use un lenguaje propio de implementación diferente a los que analizan los *OntoWrapper* existentes, se deberá desarrollar una nueva aplicación (un nuevo *OntoWrapper*) que extraiga la información de tales ontologías y la transforme en instancias de la RO.

Las **restricciones** que contempla el método propuesto son las siguientes:

1. El método *OntoMetric* contempla el uso del modelo conceptual de las ontologías de dominio, no de sus instancias. Así, el término "utilización de la ontología" que se refiere en toda la memoria es relativo al modelo conceptual de una ontología para ser usada en un sistema, y no el de los posibles conjuntos de instancias, que pueden estar situados en diferentes lugares y creados por diferentes usuarios.
2. El producto del método *OntoMetric* es una medida cuantitativa de idoneidad para cada una de las ontologías candidatas; pero también proporciona una documentación que muestra los criterios y ponderaciones seguidos por los usuarios para llegar a esas medidas. De esta forma, cuando la valoración de las ontologías haya sido realizada por varios usuarios, deberán contrastarse las documentaciones obtenidas para tomar una decisión conjunta.
3. El método no define los pasos que deben realizarse para integrar o mezclar varias ontologías, cuando el usuario considere que varias de las candidatas analizadas pueden utilizarse en su sistema. Este problema queda fuera de las pretensiones de este trabajo.
4. No se han establecido los pesos por defecto para cada categoría de proyectos basados en ontologías. Es decir, es posible que para proyectos que pertenezcan al mismo dominio, se obtengan conjuntos de pesos similares. Esto podría servir de ayuda para los usuarios que quieran valorar proyectos de un mismo dominio; pero este hito queda fuera de las pretensiones de este trabajo.
5. *OntoMetric* realiza una extracción de la información relevante de ontologías existentes, recopilada desde diversos servidores y páginas *web* de ontologías. Estos servidores y páginas *web* han sido localizados a partir de publicaciones relacionadas y búsquedas en la *web*. El método no describe cómo deben hallarse nuevos servidores, simplemente ha seleccionado los lugares *web* muy conocidos con ontologías disponibles.
6. El marco de características multinivel propuesto por el método deberá ser adaptado a las novedades conceptuales y tecnológicas que aparezcan en el futuro en el campo de las ontologías. De igual forma, deberá ser mantenida la información de las instancias sobre ontologías, lenguajes, metodologías, entornos de desarrollo y costes que se actualice en los próximos años.

Capítulo 4

Desarrollo del Método OntoMetric

4.1. INTRODUCCIÓN

El método *OntoMetric* consiste en una serie de procesos que ayudan a la selección de ontologías existentes, y proporciona una medida cuantitativa sobre la adecuación de las ontologías seleccionadas según las necesidades de un determinado sistema. En este capítulo se explicará en qué consiste el método *OntoMetric* y los procesos que lo componen. En la sección 4.2 se expone de forma esquemática el método, en la sección 4.3 se presenta el soporte tecnológico y, en las siguientes secciones, se describen en detalle las soluciones aportadas.

4.2. VISIÓN GENERAL DEL MÉTODO *ONTOMETRIC*

OntoMetric, esquematizado en la figura 4.1, consta de dos procesos principales que debe seguir secuencialmente el usuario. El producto final del método es una **medida de idoneidad para cada una de las ontologías candidatas que podrían ser utilizadas en el proyecto que se va a desarrollar**. Los dos procesos son los siguientes:

A. **Selección de ontologías**: para poder efectuar este proceso es necesario disponer de información de ontologías existentes procedentes de servidores de ontologías y de páginas *web*. Además, la información almacenada de estas ontologías debe seguir un marco conceptual (o esquema común), que será el **marco multinivel de características** que se propone en la sección 4.4. El marco se divide, en el nivel superior, en dimensiones o aspectos fundamentales que deben considerarse cuando se va a examinar una ontología. Estas dimensiones son descritas por otros aspectos más específicos que se denominan factores. Y estos, a su vez, se definen por un conjunto de características dispuestas jerárquicamente en varios niveles de detalle. En esta memoria se empleará el término **criterio** para referir, indistintamente, a dimensiones, factores, o características del marco multinivel.

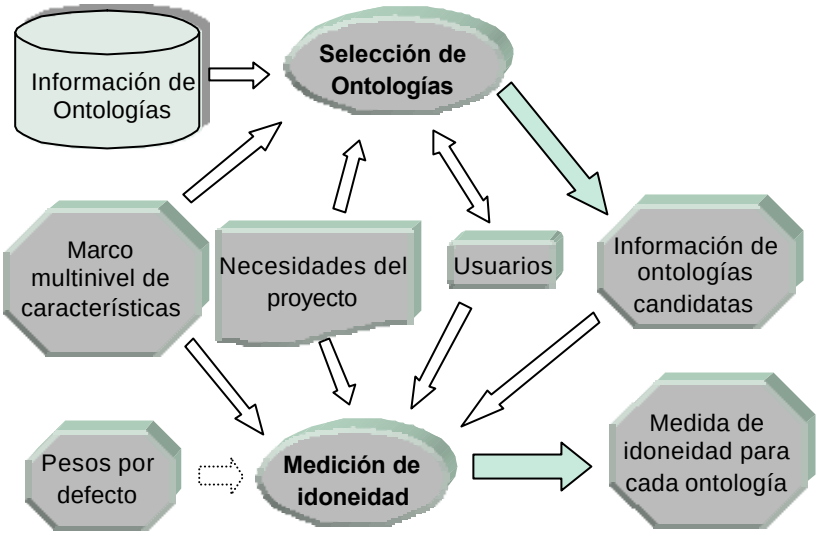


Figura 4.1. Procesos del método *OntoMetric*.

El producto de este proceso de selección será un conjunto de **ontologías candidatas**, sobre las que se hará el análisis de su idoneidad al nuevo sistema. Las entradas necesarias para realizar este proceso serán:

A.1. El marco multinivel de características.

A.2. Unos valores obligatorios en las características que serán exigidos a las ontologías examinadas. Estos valores serán indicados por los usuarios a partir de las necesidades del proyecto que se va a desarrollar.

A.3. Información sobre las ontologías existentes descritas conforme al marco multinivel de características.

B. **Medición de idoneidad**: para cada una de las ontologías candidatas obtenidas en el proceso anterior se va a asignar una medida de idoneidad en relación con el proyecto que las va a utilizar. Para obtener estas medidas, se ha adaptado el método AHP para elegir ontologías según las necesidades de un sistema. Este proceso usa como referencia el marco multinivel de características propuesto en el método, en el que se señala la importancia que tiene cada uno de los criterios según sean los objetivos y directrices del proyecto. El usuario puede utilizar un conjunto de pesos por defecto como ponderación inicial de cada criterio. Para poder realizar este proceso, se necesitan las siguientes entradas:

B.1. El marco multinivel de características sobre el que se pondera la importancia de cada uno de los criterios.

B.2. A partir de los objetivos del proyecto, se van a asignar unos pesos de importancia a

cada uno de los criterios del marco multinivel de características, obtenidos según las directrices y objetivos del proyecto. O, de forma opcional, se pueden utilizar los pesos por defecto que el método proporciona para el marco multinivel de características.

B.3. Información de las ontologías candidatas seleccionadas en el proceso anterior.

Por lo tanto, considerando las necesidades de los dos procesos, las entradas demandadas son:

- El marco multinivel de características (en A.1, B.1) que se ha desarrollado en este trabajo y que se expone en la sección 4.4.
- Valores concretos en las características del marco multinivel de características (en A.3). La forma en la que se almacenan los conocimientos para obtener estos valores se expone en la sección 4.5.
- Pesos por defecto en el marco multinivel de características (en B.2), que el usuario puede utilizar de forma opcional para valorar la idoneidad de las ontologías. La forma de obtener los pesos por defecto se describe en el apartado 4.7.1.2.2.
- La definición de los objetivos del proyecto (en A.2, B.2) que debe realizarse como una de las fases iniciales en el desarrollo de cualquier proyecto software [CSI00]. Esta es una premisa, indicada en las hipótesis de trabajo, para aplicar el método *OntoMetric*.
- Información sobre las ontologías seleccionadas como candidatas (en B.3) que la proporciona el proceso A.

A modo de resumen, los procesos propuestos por *OntoMetric* se realizan de la siguiente forma:

- El **proceso de selección de ontologías** (A). A partir de la información de las ontologías existentes, se realizará la selección indicando unos valores obligatorios en el marco multinivel de características, como se describe en la sección 4.6.
- El **proceso de asignación de medidas de idoneidad** (B) será llevado a cabo con la métrica de idoneidad de ontologías basado en el método AHP, que se verá en la sección 4.7.

4.3. SOPORTE TECNOLÓGICO PARA EL MÉTODO *ONTOMETRIC*

Se ha usado una serie de aplicaciones software para representar la información, y para asistir en la utilización del método. Algunas de estas aplicaciones han sido creadas expresamente para este fin. El esquema de la arquitectura tecnológica de *OntoMetric*, donde pueden verse las relaciones entre las aplicaciones que lo componen, se muestra en la figura 4.2. Las aplicaciones utilizadas por *OntoMetric* se definen en los dos siguientes apartados.

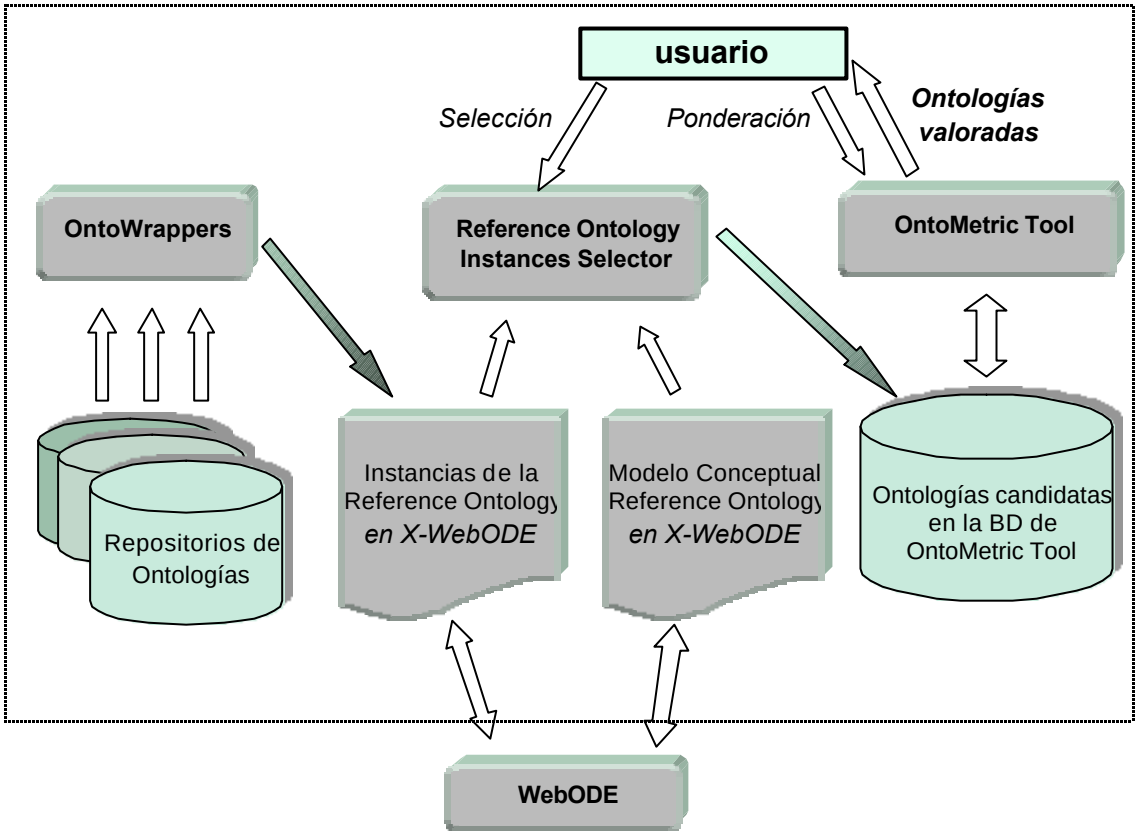


Figura 4.2. Arquitectura tecnológica de *OntoMetric*.

4.3.1. Aplicaciones para Representar el Modelo Conceptual y las Instancias de la *Reference Ontology*

OntoWrappers: se han desarrollado un conjunto de aplicaciones que extraen la información relevante relacionada con el contenido de ontologías disponibles en la *web* y en servidores de ontologías, y la transforman en instancias válidas de la dimensión "contenido" de la RO. La extracción de información se realiza desde los ficheros donde se encuentran almacenadas las ontologías, las cuales están implementadas en diferentes lenguajes.

El producto obtenido tras ejecutar los *OntoWrappers* son ficheros de texto que siguen un

formato *XML* propietario de *WebODE*, cuya *DTD* se encuentra definida en la *URL*: http://babage.dia.fi.upm.es/webode/DTD/webode_1_0.dtd. A este lenguaje de intercambio de información, a partir de ahora, se le llamará ***X-WebODE***. Los ficheros obtenidos por los *OntoWrappers*, en formato *X-WebODE*, serán importados por la aplicación *WebODE*.

WebODE [Arp01]: esta plataforma de desarrollo de ontologías, que sigue la metodología *METHONTOLOGY* ([Fer97] y [Gom98]) para la creación de ontologías, ha sido desarrollada en el Laboratorio de Inteligencia Artificial de la Universidad Politécnica de Madrid. Esta plataforma ha sido utilizada para construir el modelo conceptual de la RO. Además, se ha utilizado para incorporar las instancias de la RO en las dimensiones de "lenguajes", "metodologías", "entornos de desarrollo", y "costes de uso" de ontologías, usando su módulo de creación de instancias. Además, se ha usado su módulo de importación para incorporar instancias de la dimensión "contenido", desde los ficheros *X-WebODE* producidos por los *OntoWrappers*.

4.3.2. Aplicaciones para Asistir en la Aplicación del Método *OntoMetric*

Reference Ontology Instances Selector (**ROIS**): esta aplicación transforma directamente el modelo conceptual de la ontología RO (especificado en *X-WebODE*) en formularios *web*, para permitir al usuario realizar consultas sobre las instancias de la RO. El resultado de las consultas será el conjunto de candidatos (lenguajes, metodologías, entornos u ontologías) que satisfacen los requisitos indicados.

OntoMetric Tool (**OT**): se ha desarrollado una aplicación que asiste en los procesos de ponderación de criterios y en el cálculo de los valores de idoneidad de las ontologías candidatas. OT tiene como entrada el conjunto de ontologías candidatas obtenidas por ROIS y utiliza como esquema, para establecer las comparaciones, el marco multinivel de características de ontologías. Con esta aplicación, el usuario puede establecer los pesos de importancia de cada criterio del marco multinivel de características. O, en cambio, puede utilizar pesos por defecto para asignarlos al conjunto de criterios; de esta forma, sobre todo para el usuario poco experimentado, le pueden servir de guía de referencia. El resultado final de OT es una valoración de idoneidad de cada ontología candidata con relación a las necesidades del proyecto, indicados por el usuario.

4.4. MARCO MULTINIVEL DE CARACTERÍSTICAS PARA DESCRIBIR ONTOLOGÍAS

En esta sección se presenta un conjunto de características que deberían analizarse para elegir y comparar ontologías que van a utilizarse en un sistema. Después de examinar, en la sección 2.5. del estado de la cuestión, otras propuestas y concluir que, bajo el punto de vista de las necesidades de un proyecto, ninguno de los marcos de características encontrados sirve para decidir si una ontología es, o no, adecuada para usarla en un determinado proyecto, a continuación se presenta una taxonomía de características, también llamada **marco multinivel de características**, que proporciona el esquema para poder elegir y comparar ontologías existentes. El marco se emplea, por un lado, como esquema común de representación de la información de ontologías existentes. Por otro, sirve para que el usuario indique los requisitos necesarios que deben cumplir las ontologías que se vayan a considerar como candidatas. Finalmente, como se indicará en el apartado 4.4.1.2, es el esqueleto utilizado para formar el árbol multinivel de características utilizado en los procesos de *OntoMetric*.

4.4.1. Definición del Marco Multinivel de Características de Ontologías

La construcción del marco se ha realizado formando una taxonomía de características, agrupadas bajo el punto de vista del usuario que estudia una ontología para usarla en su proyecto. En esta sección se identifican y describen las características que han servido como base para crear, como se expone en la sección 4.4.2, una ontología en el dominio de las ontologías (la RO).

En el apartado 4.4.1.1, se definen las dimensiones principales que deben examinarse para elegir una ontología; en el apartado 4.4.1.2 se explica cómo se dispone el marco de características en un árbol multinivel de características; y, en los siguientes apartados de esta sección, se detallan las características identificadas en cada una de las dimensiones.

4.4.1.1. Las Dimensiones del Marco Multinivel de Características

El marco multinivel de características que se presenta en esta memoria posee, en el nivel superior de la taxonomía, cinco aspectos básicos sobre las ontologías, que se denominan **dimensiones**. Estas dimensiones son los aspectos principales que el usuario debe considerar para examinar una ontología para usarla en su proyecto. Estas son: el contenido de la ontología y la organización de sus contenidos, el lenguaje en el que se encuentra implementada, la metodología que se ha seguido para desarrollarla, los entornos software de desarrollo

relacionados, y el coste que va a suponer usar la ontología en un determinado proyecto.

En cada dimensión pueden aparecer características de dos tipos:

- **Características descriptivas**. Son características de índole informativa sobre algún aspecto de la dimensión, y que no se vinculan a características más generales situadas en niveles superiores. Su valor no va a repercutir en el cálculo final de la idoneidad de la ontología, pero sí podrán servir para seleccionar las ontologías candidatas, ya que podrá exigirse que, para que una ontología pueda considerarse como candidata, cumpla unos valores específicos o mínimos en algunas de estas características.
- **Características usadas para la medición de idoneidad**. Estas características servirán para obtener las medidas de idoneidad de las ontologías. Se disponen de forma jerarquizada, de manera que una característica podrá estar definida por otras más específicas, y, a partir de éstas, obtener el valor de la primera.

Como se ha mencionado, las **dimensiones** se definen a través de un conjunto de factores. Los **factores** son los elementos fundamentales que deben ser analizados para obtener el valor de las dimensiones. Estos factores se definen a través de un conjunto de **características** que permiten calcular el valor de su idoneidad. Dichas características se pueden definir, a su vez, mediante otras características, o **subcaracterísticas** más específicas, de forma que las describan con más detalle, porque se considere que deban ser analizadas con mayor profundidad. Así, el marco multinivel de características está organizado de forma taxonómica, en el que las dimensiones son descritas por factores, los factores por características, y éstas pueden estarlo por otras subcaracterísticas más específicas. Como ya se ha dicho, en esta memoria se usará el término **criterio** para referir indistintamente a cualquier elemento del marco multinivel, es decir, a las dimensiones, factores, características y subcaracterísticas de la taxonomía.

Se debe tener en cuenta que, como ya se indicó en el capítulo 3, el marco de características está sujeto a las novedades conceptuales y tecnológicas que aparezcan en el futuro en el campo de las ontologías, en el que quizás deban ser tenidos en cuenta otros factores o características que recojan las innovaciones producidas. En este sentido, el marco multinivel de características constituye un conjunto de criterios "evolutivos" que debe ser actualizado conforme a los cambios producidos.

4.4.1.2. Árbol Multinivel de Características

Las características identificadas en el marco multinivel de características pueden tomar unos

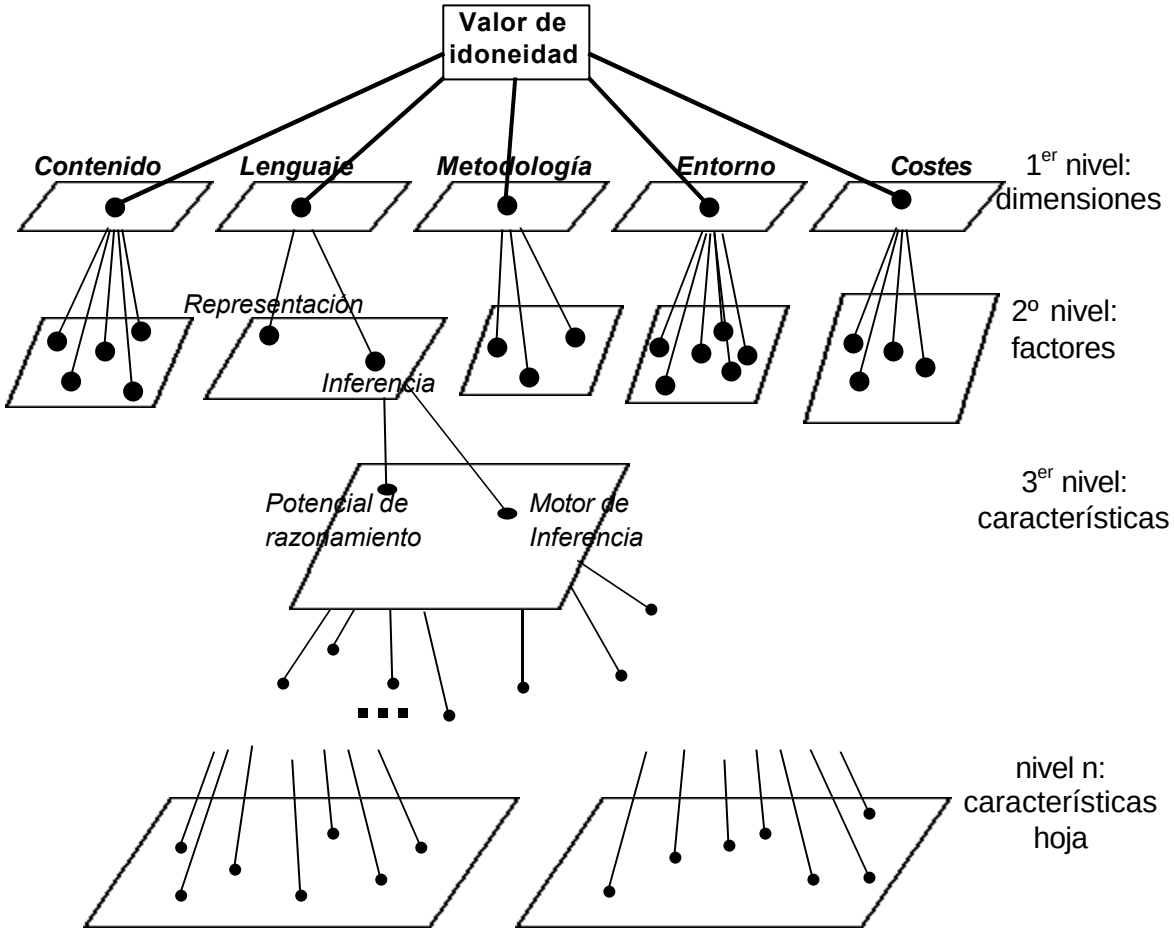


Figura 4.3. Representación del árbol multinivel de características.

valores concretos (por ser lo suficientemente específicas) para una ontología, o pueden ser valoradas a partir de otras subcaracterísticas que la describirán con mayor detalle, según el grado de definición que necesite el usuario para su análisis.

Por esta razón, se puede decir que los criterios del marco multinivel de características se han dispuesto formando un árbol jerárquico, de manera que los criterios situados en los nodos hijos describen y representan propiedades del nodo padre; y cuanto más se profundice en los niveles del árbol, con mayor detalle se describirán los nodos predecesores. Así, el usuario podrá ampliar o podar los criterios que considere oportunos, de forma que el nuevo árbol recoja las directrices establecidas para el proyecto. Los criterios forman, pues, un árbol taxonómico donde cada nodo posee, como nodos hijos, aquellos criterios que influyen en él, y que pueden usarse para determinar su valor. A este árbol se denomina **árbol multinivel de características**, que se abrevia como AMC, y puede representarse de forma gráfica como el que aparece en la figura 4.3.

Cada nivel que se descienda en el AMC indicará que se está aumentando el detalle en el análisis de esa rama. Siguiendo el ejemplo de la figura 4.3, para determinar la idoneidad de la

ontología, en el primer nivel se muestran las dimensiones que deben examinarse para valorar el nodo raíz: "idoneidad de la ontología". En la figura también aparece, como ejemplo, expandida la dimensión "lenguaje de implementación" en sus dos factores: "representación del conocimiento" y "mecanismos de inferencia". Esto indica que los valores de idoneidad que tomen estos factores servirán para hallar la idoneidad de la dimensión "lenguaje de implementación". Profundizando en el AMC hasta el detalle requerido, se llegará a las características situadas en los nodos hoja del AMC, a las que se les asignará un valor para cada ontología candidata. A partir de estos valores, de forma recursiva se podrán calcular los valores de los criterios superiores (sus nodos padre) hasta obtener el valor correspondiente del nodo raíz.

Así, a partir del AMC, y dependiendo del grado de detalle en el que se quiera realizar el análisis, el usuario podrá profundizar en los niveles del árbol de características. De esta forma, podrá examinar los aspectos más relevantes para su proyecto y que serán exigibles a las ontologías candidatas en cada uno de los criterios señalados.

4.4.1.3. Características Relacionadas con la Dimensión "Contenido de Ontologías"

En esta dimensión se consideran los términos que incluye la ontología y cómo éstos están organizados. En el proceso de valoración de las ontologías candidatas es fundamental examinar esta dimensión, para valorar qué grado de coincidencia existe entre el contenido que necesita el proyecto y el que contiene las ontologías analizadas. En este apartado se incluyen, además de las características descriptivas sobre el contenido, características que se refieren a los conceptos identificados en la ontología, la taxonomía que forma la jerarquía de conceptos, las relaciones (y funciones) existentes, y los axiomas.

En esta memoria se conviene que existe una separación entre el modelo conceptual y los conjuntos de instancias de la ontología. Como se ha mencionado en las restricciones de las hipótesis de trabajo (sección 3.4), la valoración de idoneidad se realizará respecto a los conocimientos representados en el modelo conceptual de la ontología, suponiendo que pueden, o no, existir varios conjuntos de instancias. Por esta razón, en este trabajo no se consideran las instancias como un factor que deba ser tenido en cuenta para valorar la dimensión "contenido" de la ontología.

En esta sección se describe, en primer lugar, las características descriptivas de la dimensión contenido y, en los siguientes apartados, las características usadas para la medición de idoneidad, agrupadas en los factores de conceptos, relaciones, taxonomía y axiomas.

4.4.1.3.1. Características Descriptivas Relacionadas con la Dimensión "Contenido"

Se incluyen en este apartado las características que identifican varios aspectos de la ontología candidata, siendo algunos de ellos simplemente informativos. Como ya se indicó, los valores que tomen algunas de estas características podrán ser muy significativos (como el dominio, estado de desarrollo, formalidad, etc.) y podrán descartar una ontología como candidata.

- **Nombre**: es el nombre de la ontología.
- **Descripción**: breve descripción en lenguaje natural de la ontología.
- **Servidor de la ontología**: si corresponde, la dirección URL del servidor *web* donde se encuentra la ontología. El servidor de ontologías puede contener claves de acceso a usuarios, entornos de búsqueda y explotación de ontologías, herramientas software de traducción a lenguajes de implementación, etc.
- **Dirección URL**: si corresponde, la URL desde donde se puede visualizar o descargar directamente la ontología desde la *web*. Algunas ontologías son accesibles a partir de un servidor de ontologías (como por ejemplo las situadas en el *Ontolingua Server*), y otras pueden descargarse directamente desde una dirección URL (como por ejemplo muchas de las que se construyen en el lenguaje *DAML*). La dirección URL se refiere a las ontologías que permiten esta segunda opción.
- **Estado de desarrollo**: indica la situación en la que se encuentra la ontología: acabada, en desarrollo, o si es un estudio teórico.
- **Tipo de ontología**: siguiendo la clasificación de van Heist [vHE97] puede ser una ontología de representación, de aplicación, de dominio o genérica.
- **Dominio** que representa la ontología. Puede ser usual que el usuario utilice el valor que aparezca en esta característica como una primera aproximación en la búsqueda de ontologías candidatas.
- **Generalidad**: relativo a si los términos que representan son de ámbito general o de un dominio específico. Si son de ámbito general, sus términos pueden formar parte una ontología de nivel superior (*upperlevel ontology*).
- **Propósito**: para el que fue construida la ontología. Si la ontología examinada ha sido construida con el mismo propósito que el proyecto, posiblemente coincidirán en mayor medida los términos y su organización taxonómica. Sin embargo, si el propósito no coincide, no implica que la reutilización no pueda llevarse a cabo. Por ejemplo, esta característica puede tomar los valores de: procesamiento de lenguaje natural, simulación, enseñanza, de propósito comercial, investigación teórica, consensuar conocimiento, etc.

- **Formalidad**: relativo al grado de formalidad empleado en la especificación de los términos. Puede ser: altamente informal, semi-informal, semi-formal, o rigurosamente formal [Usc96b].
- **Ontologías incluidas**: lista de ontologías que la ontología considerada incluye.
- **Paradigmas de formalización** usados para representar los conocimientos en la ontología: basado en marcos, lógica descriptiva, lógica clásica, no-clásica, red semántica, grafos conceptuales, esquema de representación híbrido (marcos y lógica), etc.
- **Desarrollador de la ontología**: indica las personas o entidades que han construido la ontología.
- **Fuentes de conocimientos**: contiene las fuentes de conocimientos utilizadas para adquirir los conocimientos. Pueden ser expertos del dominio, libros, bases de datos, otras ontologías, etc.
- **Estrategia para identificar conceptos**: se refiere a cómo fue construida la ontología. Puede realizarse desde los conceptos generales a los más concretos, desde los concretos a los generales, o utilizando una aproximación media [Usc96b].
- **Fecha de publicación**: fecha en la que la ontología se ha dado a conocer a la comunidad de usuarios de ontologías. Esta información puede ser relevante cuando la ontología trate de un dominio con continuas actualizaciones.
- **Fecha de la última modificación**: contrastada con la fecha de publicación, dará idea de lo estable o cambiante que es la ontología.
- **Proyectos significativos que usan la ontología**: será más fiable si la ontología ha sido utilizada en otros proyectos. Deberá examinarse si los proyectos son del mismo dominio y si tienen el mismo alcance que los del proyecto en estudio.

4.4.1.3.2. Características de la Dimensión "Contenido" Usadas para la Medición de Idoneidad

4.4.1.3.2.1. Características que Influyen en el Factor "Conceptos"

En este apartado se exponen las características relacionadas con los conceptos de la ontología, que servirán para obtener el valor de idoneidad del factor "conceptos". Se han escogido de manera que el usuario pueda analizar el grado de coincidencia de los conceptos de la ontología y la forma en la que están especificados, con lo que se necesita para su proyecto. Las

características que deben ser analizadas son las siguientes:

- **Los conceptos esenciales para el proyecto están en la ontología**: el usuario debe buscar en la ontología candidata los conceptos que considere fundamentales para su proyecto, es decir, aquellos que está seguro de que va a utilizar en su sistema. Se debe tener en cuenta que, cuando se hace una búsqueda por un nombre para un determinado concepto, en la ontología pueden aparecer sinónimos. Por lo tanto, para valorar esta característica, se debe examinar no sólo el nombre, sino también los atributos que el concepto posee y la descripción que se hace en lenguaje natural del concepto. El valor que tomará esta característica indicará el grado de coincidencia de los conceptos de la ontología con las necesidades del proyecto.
- **Los conceptos esenciales para el proyecto están en los niveles superiores de la taxonomía**: si los conceptos que se van a necesitar para el proyecto están en los niveles superiores de la ontología, se puede decir que se podrán reutilizar más términos que si aparecen en niveles inferiores de otra taxonomía. En este último caso, se deberá realizar un esfuerzo adicional en especializar los conceptos de la ontología a las necesidades del proyecto.
- **Los conceptos están descritos convenientemente en lenguaje natural**: si la definición informal de los conceptos encontrados en la ontología es escueta y ambigua se necesitará realizar un esfuerzo adicional para entender su definición formal. Definiciones en lenguaje natural poco precisas o ambiguas dificultarán el uso de la ontología en la aplicación.
- **La especificación formal de los conceptos coincide con su descripción en lenguaje natural**: si la definición formal de los conceptos dada por la declaración de sus atributos, relaciones, axiomas, etc. no se corresponde con la descripción en lenguaje natural del concepto, indicará que la ontología no está construida correctamente [Gom99c].
- **Los atributos describen de forma precisa a los conceptos**: se debe comprobar si los atributos de los conceptos que existen en la ontología son los que necesita el proyecto que se va a desarrollar ya que, si no es el caso, se necesitará añadir o modificar estos atributos, con el consiguiente incremento de esfuerzo. Además de la existencia de estos atributos, se debe examinar si están bien definidos y si sus facetas se corresponden con las que se buscan para el proyecto.
- **El número de conceptos representados en la ontología**: el usuario debe estimar si el tamaño de la ontología, relativo al número de conceptos que contiene, es el adecuado para el proyecto. Un número pequeño respecto a las necesidades del proyecto implicará realizar un esfuerzo para añadir estos conceptos; un número grande implicará una ontología menos manejable y, quizás, un proceso de selección de conceptos.

4.4.1.3.2.2. Características que Influyen en el Factor "Relaciones"

Como en el caso de los conceptos, el usuario debe establecer en qué grado las relaciones existentes satisfacen las necesidades del proyecto, y si la forma en la que están especificadas es la que se necesita. Las siguientes características servirán para obtener el valor de idoneidad del factor "relaciones".

- **Las relaciones esenciales para el proyecto están en la ontología**: el usuario debe buscar en la ontología candidata las relaciones que considere fundamentales. Al igual que con los conceptos, para identificar la relación debe examinarse no sólo el nombre que le ha asignado el desarrollador de la ontología, sino la descripción que se hace en lenguaje natural de la relación, y su definición formal.
- **Los conceptos se encuentran relacionados como se necesitan en el sistema**: a pesar de que la ontología tenga definida una relación, puede ser que la ontología no proporcione la relación tal cual se necesite en el proyecto, es decir, que la relación esté definida entre conceptos diferentes a los que se necesite en el proyecto.
- **Las relaciones están descritas convenientemente en lenguaje natural**: si las definiciones informales de las relaciones encontradas en la ontología es demasiado escueta o ambigua, se necesitará realizar un esfuerzo adicional para entender su definición formal.
- **La aridad de las relaciones coincide con la que necesita el proyecto**. Se deberá comprobar si hay coincidencia en el número de conceptos que asocia la relación.
- **Las relaciones tienen especificadas sus propiedades formales**: al igual que ocurre con la aridad, deberán examinarse las propiedades formales de las relaciones, y evaluar su adecuación con las necesidades del proyecto. Siendo R una relación, y $x_1,...,x_n$, conceptos, las propiedades matemáticas clásicas que se pueden definir en las relaciones [Sta00] son:
  - Reflexividad: $\forall\, x_1,\ R(x_1, x_1)$
  - Irreflexividad: $\forall\, x_1 :\ \neg R(x_1, x_1)$
  - Simetría: $\forall\, x_1, x_2, R(x_1, x_2) \rightarrow R(x_2, x_1)$
  - Asimetría: $\forall\, x_1, x_2, R(x_1, x_2) \rightarrow \neg R(x_2, x_1)$
  - Antisimetría: $\forall\, x_1, x_2, R(x_1, x_2) \wedge R(x_2, x_1) \rightarrow x_1 = x_2$
  - Transitividad: $\forall\, x_1, x_2, x_3, R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow R(x_1, x_3)$
  - Intransitividad: $\forall\, x_1, x_2, x_3, R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow \neg R(x_1, x_3)$

  Este podría ser un ejemplo típico en el que el usuario decida detallar esta característica con el estudio de cada una de las propiedades formales, que serían consideradas como subcaracterísticas de la característica "las relaciones tienen especificadas sus propiedades

formales".

- **El número de relaciones representadas en la ontología**: el usuario debe estimar si el número de relaciones que contiene la ontología, es el adecuado para el proyecto. Un número pequeño respecto a las necesidades del proyecto implicará realizar un esfuerzo para añadir las relaciones; un número grande implicará una ontología menos manejable y, quizás, un proceso de selección de relaciones útiles.

4.4.1.3.2.3. Características que Influyen en el Factor "Taxonomía de Conceptos"

El usuario debe analizar cómo están organizados los conceptos en la ontología, y si la taxonomía que la ontología proporciona es la adecuada para el proyecto. Las características que deben examinarse para valorar este factor son:

- **Los conceptos son clasificados desde varias perspectivas**: puede ser que, para el proyecto, se necesite que la clasificación de los conceptos no sea única. Debe examinarse si la ontología contiene varias relaciones del tipo *subclase_de* en un mismo concepto, y si esta clasificación es la que necesita el sistema.
- **La relación *no_subclase_de* es usada convenientemente para las necesidades del sistema**: de igual forma, para el proyecto puede ser interesante que se especifique esta relación, normalmente usada para romper una herencia entre conceptos.
- **La relación de partición disjunta es usada convenientemente para las necesidades del sistema**: una partición disjunta entre clases indica que dichas clases no pueden tener instancias ni subclases comunes. Las particiones disjuntas suelen ser importantes en procesos de clasificaciones automáticas.
- **La relación de partición exhaustiva es usada convenientemente para las necesidades del sistema**: una partición exhaustiva entre clases es una partición disjunta, a la que se le añade el requisito de completitud. Es decir, el conjunto de clases que forman la partición define de forma completa a la clase padre. Nótese que en la partición disjunta este requisito de completitud no se añade. Debe examinarse si tanto las anteriores particiones disjuntas como las particiones exhaustivas están definidas tal y como se necesitan para el proyecto.
- **La profundidad máxima en la jerarquía de conceptos**: definida como el mayor camino existente siguiendo las relaciones de herencia que puede alcanzar la taxonomía. Para obtener una apreciación válida del grado de detalle de la ontología, debe analizarse conjuntamente con el número de conceptos que contiene la ontología.
- **La media de hijos por concepto**: el número medio de subclases por cada clase dará una idea del nivel de detalle en el que está definida la taxonomía de la ontología.

4.4.1.3.2.4. Características que Influyen en el Factor "Axiomas"

Los axiomas son utilizados para restringir los valores de los atributos de las instancias e instancias de relaciones, mantener la consistencia de la ontología, y para hacer deducciones. Para obtener el valor de idoneidad del factor "axiomas", deberán analizarse las siguientes características:

- **Los axiomas pueden utilizarse para hacer deducciones respondiendo a consultas**. Algunos axiomas pueden estar definidos de forma que pueden ser utilizados en el proyecto para inferir conocimientos no explícitos en la base de conocimientos de la aplicación (a estos axiomas se les denominan reglas [Cor00]). De esta forma, el usuario deberá analizar el potencial de estos axiomas con relación a las necesidades de su proyecto para deducir nuevos conocimientos, y para realizar búsquedas de conocimientos no explícitos.
- **Los axiomas pueden utilizarse para completar valores en los atributos de instancia** de las instancias de la base de conocimientos del nuevo sistema: al crear una nueva instancia en el sistema, algunos valores de los atributos de instancia podrán ser deducidos utilizando los axiomas definidos en la ontología. También pueden utilizarse para deducir las instancias de relaciones.
- **Los axiomas pueden utilizarse para verificar la consistencia**: los axiomas pueden utilizarse para verificar los valores que puedan tomar los atributos de los conceptos, o las relaciones permitidas entre conceptos que se definan en el nuevo sistema.
- **Los axiomas están definidos como elementos independientes en la ontología**: los axiomas definidos fuera de la declaración de los términos de la ontología dan mayor facilidad de comprensión y de modificación, ya que su definición no dependerá de los cambios que se hagan en los otros términos de la ontología.
- **El número de axiomas de la ontología**: la cantidad de axiomas definidos puede dar una idea del potencial de deducción especificado en la ontología, y de la capacidad de mantener la consistencia.

Como resumen del marco de características propuesto para la dimensión "contenido", en la tabla 4.1 se muestran las características, agrupadas en sus correspondientes factores. Se indican también los valores que puede tomar cada característica, la cardinalidad mínima y máxima, y el autor que la identifica, junto con su referencia bibliográfica. Se debe señalar que el tipo de valor en las características usadas para la medición de idoneidad toma los valores: *Muy Bajo*, *Bajo*, *Medio*, *Alto*, *Muy Alto*; estos valores serán asignados por los usuarios conforme a la idoneidad de la ontología en esa característica para el proyecto que se está valorando.

Cód	Nombre de la característica	Tipo de valor	Cardinalidad	Autor	Referencia
Características descriptivas					
C01	Nombre de la ontología	Texto	(1,1)		
C02	Descripción	Texto	(0,1)		
C03	Servidor de la ontología	URL	(0,n)		
C04	Dirección URL	URL	(0,n)		
C05	Estado de desarrollo	(Estudio, En desarrollo, Finalizada)	(0,1)	Uschold	[Usc98a]
C06	Tipo de ontología	(Aplicación, Dominio, Genérica, de Representación)	(0,n)	van Heist et al.	[vHe97]
C07	Dominio	Texto	(0,n)	Uschold y Grüninger	[Usc96a]
C08	Generalidad	(MuyBaja, Baja, Media, Alta, MuyAlta)	(0,1)	Noy y Hafner	[Noy97]
C09	Propósito	Texto	(0,n)	Uschold y Grüninger	[Usc96a]
C010	Formalidad	(AltamenteInformal, Semi-informal, Semi-formal, RigurosamenteFormal)	(0,1)	Uschold y Grüninger	[Usc96a]
C011	Ontologías incluidas	Texto	(0,n)		
C012	Paradigmas de formalización	(Lógica clásica, Lógica no clásica, Lógica monótona, Lógica no monótona, Lógica descriptiva, Basado en Marcos, Redes semántica, etc.)	(0,n)		
C013	Desarrollador de la ontología	Texto	(0,n)		
C014	Fuentes de conocimientos	Texto	(0,1)		
C015	Estrategia para identificar conceptos	(De arriba a abajo, De abajo a arriba, Aproximación media)	(0,1)	Uschold y Grüninger	[Usc96b]
C016	Fecha de publicación	Fecha	(0,1)		
C017	Fecha de la última modificación	Fecha	(0,1)		
C018	Proyectos significativos que usan la ontología	Texto	(0,n)	Noy y Hafner (*aplicaciones*)	[Noy97]
C1	**FACTOR: Conceptos**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Gruber	[Gru95a]
C11	Los conceptos esenciales para el proyecto están en la ontología	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	-parcialmente Noy y Hafner (*términos representados*) -parcialmente Hovy (*cobertura sobre el dominio como característica de terminología*)	[Noy97], [Hov97]
C12	Los conceptos esenciales para el proyecto están en los niveles superiores de la ontología	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	parcialmente Noy y Hafner (*división en el nivel superior*)	[Noy97]
C13	Los conceptos están descritos convenientemente en lenguaje natural	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C14	La especificación formal de los conceptos coincide con su descripción en lenguaje natural	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Gómez-Pérez (*evaluación*)	[Gom99c]
C15	Los atributos describen de forma precisa a los conceptos	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C16	Es adecuado el número de conceptos representados en la ontología	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	-parcialmente Noy y Hafner (*tamaño*) -parcialmente Uschold (*tamaño*) -parcialmente Hovy (*tamaño como característica de terminología*)	[Noy97], [Usc98a], [Hov97]
C2	**FACTOR: Relaciones**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Gruber	[Gru95a]
C21	Las relaciones esenciales para el proyecto están en la ontología	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C22	Los conceptos se encuentran relacionados como se necesitan para el sistema	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C23	Las relaciones están descritas convenientemente en lenguaje natural	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C24	La aridad de las relaciones coincide con la que necesita el proyecto	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C25	Las relaciones tienen especificadas sus propiedades formales	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Staab y Maedche	[Sta00]
C26	Es adecuado el número de relaciones representadas en la ontología	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	-parcialmente Noy y Hafner (*tamaño*) -parcialmente Uschold (*tamaño*) -parcialmente Hovy (*tamaño como característica de terminología*)	[Noy97], [Usc98a], [Hov97]

Tabla 4.1. Características relacionadas con la dimensión "contenido" de las ontologías **(1/2)**.

Cód	Nombre de la característica	Tipo de valor	Cardi-nalidad	Autor	Refe-rencia
C3	**FACTOR: Taxonomía**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Noy y Hafner	[Noy97]
C31	La clasificación de los conceptos desde varias perspectivas es la que se necesita	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Noy y Hafner (*varias taxonomías*)	[Noy97]
C32	La relación *no_subclase_de* es usada convenientemente para las necesidades del sistema	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C33	La relación de partición disjunta es usada convenientemente para las necesidades del sistema	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C34	La relación de partición exhaustiva es usada convenientemente para las necesidades del sistema	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C35	La profundidad máxima en la jerarquía de conceptos es la que se necesita	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	-Hovy (*profundidad*) -parcialmente Noy y Hafner (*enmarañada*)	[Hov97], [Noy97]
C36	La media de hijos por concepto es la que se necesita	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	-Hovy (*ramificación*) -parcialmente Noy y Hafner (*enmarañada*)	[Hov97], [Noy97]
C4	**FACTOR: Axiomas**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Gruber	[Gru95a]
C41	Los axiomas pueden utilizarse para hacer deducciones respondiendo a consultas	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C42	Los axiomas pueden utilizarse para completar valores en los atributos de instancia e instancias de relaciones	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C43	Los axiomas pueden utilizarse para verificar la consistencia de los términos de la ontología	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C44	Los axiomas están definidos como elementos independientes	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
C45	Es adecuado el número de axiomas de la ontología	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	-parcialmente Noy y Hafner (*tamaño*) -parcialmente Uschold (*tamaño*) -Hovy (*número de axiomas*)	[Noy97], [Usc98a], [Hov97]

Tabla 4.1. Características relacionadas con la dimensión "contenido" de las ontologías **(2/2)**.

4.4.1.4. Características Relacionadas con la Dimensión "Lenguajes de Implementación de Ontologías"

El lenguaje en el que se encuentra implementada la ontología, o en el que puede estarlo por medio de traductores, constituye también uno de los principales criterios que deben ser examinados para seleccionar una ontología. Por un lado, si para el proyecto van a necesitarse métodos de resolución de problemas que realicen inferencias con los conocimientos almacenados, es fundamental que el lenguaje tenga las capacidades deductivas requeridas. Además, si se necesitan añadir nuevos términos a la ontología o aumentar la granularidad de los existentes, el usuario se debe asegurarse que el lenguaje de implementación tiene ese potencial expresivo. La metodología *CommonKADS* [Sch99] identifica estos dos aspectos relacionados con los conocimientos: la representación del conocimiento y los mecanismos de inferencia. Esta división es la que han seguido Corcho y Gómez-Pérez [Cor00] para elaborar su marco de trabajo sobre características de lenguajes, que aparece descrito en el apartado 2.5.2. El marco de Corcho y Gómez-Pérez es el que se ha seguido en esta memoria para la dimensión "lenguaje de implementación" con alguna pequeña variación en su taxonomía y añadiendo algunas

características. Por esta razón, los dos factores de esta dimensión son la capacidad de representación del conocimiento en el lenguaje y sus mecanismos de inferencia.

Se describen a continuación las características descriptivas identificadas para la dimensión "lenguaje de implementación", y posteriormente se describen las características del lenguaje usadas para la medición de su idoneidad, señalando las diferencias con el marco propuesto en [Cor00].

4.4.1.4.1. Características Descriptivas Relacionadas con la Dimensión "Lenguaje"

Se incluyen en este apartado las características que identifican varios aspectos informativos sobre los lenguajes de implementación de ontologías. Estas características, que aparecen en la tabla 4.2, no se encuentran en el marco de Corcho y Gómez-Pérez, y se han añadido a esta dimensión porque el usuario, en la búsqueda de lenguajes candidatos, puede solicitar esta información o puede especificar unos valores concretos en estas propiedades.

- **Nombre**: nombre del lenguaje.
- **Tipo de lenguaje**: referido a si el lenguaje es de tipo "clásico" o está diseñado para realizar marcado en páginas *web*.
- **Formalismos de representación** en el que se basa el lenguaje. Se refiere a si el lenguaje se basa en alguno o varios de los formalismos usados para representar el conocimiento: lógica clásica, lógica no clásica, lógica monótona, lógica no monótona, lógica descriptiva, marcos, redes semánticas, etc.
- **Calidad de los manuales de ayuda**: indica la calidad de los manuales existentes sobre el lenguaje, si existen ejemplos de ontologías implementadas en el lenguaje, etc.
- **El lenguaje puede ser utilizado de forma independiente de los entornos de desarrollo de la ontología:** se refiere a si una ontología implementada en un lenguaje puede ser utilizada sin necesidad de usar un entorno software específico. Esta característica puede tomar varios valores, por darse diferentes situaciones: a) la ontología escrita en un determinado lenguaje necesita de un entorno software "propietario" para modificar sus términos; b) la ontología escrita en un determinado lenguaje necesita de un entorno software o de módulos software determinados para acceder a sus términos; y c) la ontología escrita en un determinado lenguaje puede ser utilizada directamente sin el entorno, una vez que se ha exportado desde el entorno software que se utilizó para su construcción.
- **URL con información sobre el lenguaje.**
- **Desarrollador del lenguaje.**

- **Código de la última versión del lenguaje.**
- **Fecha de la última versión:** se refiere a la fecha en la que fue publicada la última versión del lenguaje.

Cód	Nombre de la característica	Tipo de valor	Cardinalidad	Autor	Referencia
Características descriptivas					
L01	Nombre del lenguaje	Texto	(1,1)		
L02	Tipo de lenguaje	(Clásico, Marcado *web*)	(0,1)		
L03	Formalismos de representación	(Lógica clásica, Lógica no clásica, Lógica monótona, Lógica no monótona, Lógica descriptiva, Marcos, Redes semántica, etc.)	(0,n)		
L04	Calidad de los manuales de ayuda	(MuyBaja, Baja, Media, Alta, MuyAlta)	(0,1)		
L05	Permite el uso independiente del entorno	(Necesita entorno para edición, Necesita entorno para acceso, No necesita entornos para uso)	(0,n)		
L06	URL de información	URL	(0,n)		
L07	Desarrollador del lenguaje	Texto	(0,n)		
L08	Versión del lenguaje	Texto	(0,1)		
L09	Fecha de la última versión	Fecha	(0,1)		

Tabla 4.2. Características descriptivas sobre lenguajes de implementación.

4.4.1.4.2. Características de la Dimensión "Lenguaje" Usadas para la Medición de Idoneidad

Como ya se ha indicado, el marco de características que se va a seguir para medir la idoneidad del lenguaje se basa, casi en su totalidad, en la propuesta de Corcho y Gómez-Pérez, y es el que aparece en la tabla 4.3.

Aunque, como se puede ver en la tabla 4.3, la mayor parte de las características que utiliza *OntoMetric* para valorar la idoneidad del lenguaje son las propuestas por Corcho y Gómez-Pérez, se ha modificado la disposición de las características y se han identificado algunas más.

Diferencias en la taxonomía del marco. Aunque el marco de Corcho y Gómez-Pérez podría haberse usado tal y como lo proponen los autores, se ha considerado hacer algunos cambios para facilitar los procesos de comparación de factores y características. En concreto:

- Se han agrupado en una misma característica, características relacionadas con conceptos, instancias, hechos y afirmaciones.

Cód	Nombre de la característica	Tipo de valor	Cardi-nalidad	Autor	Refe-rencia
L1	**FACTOR: Representación del Conocimiento**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Schreiber et al.	[Sch99]
L11	**Característica: Conceptos/ Instancias/ Hechos/ Afirmaciones**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Corcho y Gómez	[Cor00]
L111	Permite instancias	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L112	Contiene metaclases	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L113	Permite definir clases sin usar metaclases	(No soportado, Soportado)	(1,1)		
L114	Permite hechos	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L115	Permite afirmaciones	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L12	**Característica: Atributos**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
L121	Permite atributos de clase	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L122	Permite atributos de instancia	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L123	Permite atributos locales	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L124	Permite atributos globales	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L125	Permite atributos polimórficos	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L126	Permite definir excepciones en los atributos	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L13	**Característica: Facetas**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
L131	Permite atributos por defecto	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L132	Permite definir el tipo	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L133	Permite definir la cardinalidad	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L134	Permite definir conocimiento procedural	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L135	Permite definir nuevas facetas	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L14	**Característica: Relaciones**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Corcho y Gómez	[Cor00]
L141	Permite el uso de funciones	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L142	Relaciones n-arias	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L143	Permite definir nuevas relaciones ad-hoc	(No soportado, Soportado)	(1,1)		
L144	Tipos de conceptos restringido	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L145	Restricciones de integridad	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L146	Definiciones operacionales	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L147	Permite declarar propiedades matemáticas en relaciones (inversa, reflexiva, irreflexiva, simétrica, asimétrica, antisimétrica, transitiva, intransitiva)	(No soportado, Soportado)	(1,1)		
L15	**Característica: Taxonomía**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Corcho y Gómez	[Cor00]
L151	Relación Subclase_de	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L152	Relación No-Subclase_de	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L153	Descomposiciones exhaustivas	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L154	Descomposiciones disjuntas	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L155	Una clase puede ser subclase de varias clases	(No soportado, Soportado)	(1,1)		
L156	Una instancia puede ser instancia de varias clases	(No soportado, Soportado)	(1,1)		
L16	**Característica: Axiomas**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Corcho y Gómez	[Cor00]
L161	Permite axiomas asociados a términos	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L162	Permite axiomas independientes	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L163	Lógica de primer orden	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L164	Lógica de segundo orden	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L17	**Característica: Reglas de producción**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Corcho y Gómez	[Cor00]
L171	Premisas conjuntivas	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L172	Premisas disyuntivas	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L173	Definición de prioridad	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L174	Valores de verdad en consecuente	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L175	Definir mecanismos de encadenamiento	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L176	Definir procedimientos para actualizar la BC	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L2	**FACTOR: Mecanismos de Inferencia**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Schreiber et al.	[Sch99]
L21	**Característica: Potencial de Razonamiento**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
L211	Permite herencia múltiple	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L212	Permite razonamiento monótono	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L213	Permite razonamiento no mo nótono	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L214	Maneja excepciones en la herencia	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L215	Los axiomas pueden ser usados para mantener la consistencia	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L216	Ejecuta procedimientos	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L217	Las reglas de producción tienen asociados mecanismos de encadenamiento	(No soportado, Soportado)	(1,1)		
L22	**Característica: Motor de Inferencia**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
L221	MI completo y correcto	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L222	MI realiza clasificaciones automáticas	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L223	MI maneja excepciones	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L224	MI maneja herencia múltiple	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]
L225	Se puede implementar un nuevo MI diferente al existente	(No soportado, Soportado)	(1,1)	Corcho y Gómez	[Cor00]

Tabla 4.3. Características usadas para valorar la idoneidad del lenguaje.

- De las características relacionadas con conceptos propuestas por Corcho y Gómez-Pérez, se ha considerado que deben agruparse en características separadas, las características relacionadas con los atributos de conceptos, y con las facetas de atributos.
- Respecto al "mecanismo de inferencia" del lenguaje, se ha dividido en dos criterios: por un lado, el "potencial de razonamiento" que indica las capacidades que tiene el lenguaje para poder realizar deducciones con el conocimiento almacenado; y por otro, las propiedades que posee concretamente el motor de inferencia asociado al lenguaje.

Diferencias en las características: aunque no existen grandes diferencias con las características propuestas por Corcho y Gómez-Pérez, se han añadido las siguientes:

- Se ha añadido la característica: "Las clases pueden ser definidas sin usar metaclases". Esta característica se refiere a que no es necesario utilizar una metaclase definida en el lenguaje para crear una clase nueva.
- Se ha añadido la característica: "El lenguaje permite definir nuevas relaciones ad-hoc". Esta característica indica que el lenguaje permite crear nuevas relaciones entre conceptos, además de las ya definidas en el lenguaje.
- Se ha añadido una característica que recoge la posibilidad de definir propiedades matemáticas que pueden ser exigidas a las relaciones: se refiere a si el lenguaje permite indicar que una relación es inversa de otra, si se puede indicar que una relación tiene la propiedad reflexiva, irreflexiva, simétrica, asimétrica, antisimétrica, transitiva o intransitiva (estos conceptos están definidos en el apartado 4.4.1.3.2).
- En la característica "taxonomía", se han añadido características para indicar que una clase puede ser subclase de varias clases, y que una instancia puede serlo de varias clases.
- Se ha añadido la característica: "Las reglas de producción pueden tener asignada la prioridad de selección". Esta característica indica que el lenguaje permite asignar una prioridad para escoger una regla de producción frente a otras.

4.4.1.5. Características Relacionadas con la Dimensión "Metodologías de Desarrollo de Ontologías"

El desarrollo de ontologías debe ser realizado mediante metodologías adecuadas, diseñadas para este fin, del mismo modo que se siguen metodologías específicas para el desarrollo de cualquier componente software. Una ontología desarrollada con una metodología será más fácil de

modificar que otra desarrollada sin ella. Por el contrario, una metodología con actividades difíciles de seguir o mal definidas puede retrasar considerablemente el proceso de adaptación de la ontología al nuevo sistema.

Para definir esta dimensión, se han tomado como referencia las características recogidas por Fernández en [Fer99b] y [Fer00], que aparecen en la sección 2.5.3 de este trabajo. En ambas publicaciones se muestran estas características de forma lineal, sin ninguna organización taxonómica. Además de las características descriptivas usadas para seleccionar metodologías, para ayudar al usuario a establecer los factores que permitirán valorar la idoneidad de la metodología, se han agrupado estas características (y algunas nuevas) en los factores indicados en el apartado 4.4.1.5.2. Un resumen de todas las características se muestra en la tabla 4.4.

4.4.1.5.1. Características Descriptivas Relacionadas con la Dimensión "Metodología"

Todas las características descriptivas que se incluyen en el marco multinivel de características (y que se muestran en la tabla 4.4) son las identificadas por Fernández [Fer99b], y se definieron en la sección 2.5.3.

4.4.1.5.2. Características de la Dimensión "Metodología" Usadas para la Medición de Idoneidad

4.4.1.5.2.1. Características que Influyen en el Factor "Precisión de la Metodología"

Las características incluidas en este factor indican la exactitud en la definición de las actividades, técnicas, personal involucrado, y productos de salida identificados en la metodología. Son las siguientes:

- **Delimitación de fases**: indica si están bien establecidos y definidos los procesos que se deben realizar en cada una de las fases de desarrollo.
- **Especificación de las actividades**: indica si están bien detalladas las actividades que deben llevarse a cabo en cada una de las fases de desarrollo.
- **Especificación del personal**: indica si están bien definidas las personas que deben intervenir en cada una de las fases de desarrollo.
- **Especificación de las técnicas**: indica si están bien identificadas las técnicas que deben emplearse.
- **Especificación de los productos**: indica si están descritos los formatos y el contenido de los productos que se obtendrán después de la ejecución de cada una de las fases.

4.4.1.5.2.2. Características que Influyen en el Factor "Usabilidad de la Metodología"

Este factor se define como la facilidad de comprender y aplicar las actividades y técnicas propuestas por la metodología. Las características que deben analizarse para determinar su valor son:

- **Claridad de la descripción de las actividades y técnicas**: indica si es fácilmente comprensible la forma en la que se han detallado las actividades y técnicas que deben realizarse en cada fase.
- **Calidad de los manuales de ayuda**: se refiere a la calidad del material de ayuda disponible para aprender a utilizar la metodología.
- **Existencia de ejemplos de aplicación**: indica si existen ejemplos adecuados que ayuden a comprender las técnicas y actividades de la metodología.

4.4.1.5.2.3. Características que Influyen en el Factor "Madurez de la Metodología"

En este factor se determina si la metodología es aceptada y seguida por los usuarios que desarrollan ontologías. Las características que deben ser analizadas son:

- **Número de ontologías desarrolladas con la metodología**: indica el número de ontologías desarrolladas siguiendo la metodología.
- **Número de dominios diferentes**: indica el número de dominios de ontologías diferentes en los que se ha empleado la metodología.
- **Importancia de las ontologías desarrolladas**: indica la importancia de las ontologías que se han desarrollado empleando la metodología.

Cód	Nombre de la característica	Tipo de valor	Cardi-nalidad	Autor	Refe-rencia
Características descriptivas					
M01	Nombre de la metodología	Texto	(1,1)		
M02	Descripción breve de la metodología	Texto	(0,1)		
M03	Herencia de la Ingeniería del Conocimiento	(MuyBaja, Baja, Media, Alta, MuyAlta)	(0,1)	Fernández	[Fer99b]
M04	Pasos recomendados para construir ontologías	Texto	(0,1)	Fernández	[Fer99b]
M05	Recomendaciones para la formalización del conocimiento	(Lógica clásica, Lógica no clásica, Lógica monótona, Lógica no monótona, Lógica descriptiva, Basado en Marcos, Redes semántica, etc.)	(0,n)	Fernández	[Fer99b]
M06	Estrategia para construir ontologías	(Dependiente de la aplicación, Semidependiente de la aplicación, independiente de la aplicación)	(0,1)	Fernández	[Fer99b]
M07	Estrategia para identificar conceptos	(De arriba a abajo, De abajo a arriba, Aproximación media)	(0,1)	Fernández	[Fer99b]
M08	Ciclo de vida recomendado	Texto	(0,1)	Fernández	[Fer99b]
M09	Actividades recomendadas en el ciclo de vida	Texto	(0,1)	Fernández	[Fer99b]
M010	Técnicas recomendadas	Texto	(0,1)	Fernández	[Fer99b]
M011	Soporte tecnológico	Texto	(0,1)	Fernández	[Fer99b]
M012	Ontologías desarrolladas con la metodología	Texto	(0,n)	Fernández	[Fer99b]
M1	**FACTOR: Precisión**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
M11	Delimitación de fases	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		
M12	Especificación de las actividades	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)	parcialmente Fernández en las características de *procesos orientados al desarrollo*	[Fer99b]
M13	Especificación del personal	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)	parcialmente Fernández en las características de *procesos orientados al desarrollo*	[Fer99b]
M14	Especificación de las técnicas	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)	parcialmente Fernández en las características de *procesos orientados al desarrollo*	[Fer99b]
M15	Especificación de los productos	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)	parcialmente Fernández en las características de *procesos orientados al desarrollo*	[Fer99b]
M2	**FACTOR: Usabilidad**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
M21	Claridad de la descripción de las actividades y técnicas	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		
M22	Calidad de los manuales de ayuda	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		
M23	Existencia de ejemplos de aplicación de la metodología	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		
M3	**FACTOR: Madurez**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
M31	Número de ontologías desarrolladas	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
M32	Número de dominios diferentes	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
M33	Importancia de las ontologías desarrollados	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		

Tabla 4.4. Características relacionadas con la dimensión "metodología de desarrollo de ontologías".

4.4.1.6. Características Relacionadas con la Dimensión "Entornos de Desarrollo de Ontologías"

Uno de los procesos que más tiempo consume al intentar usar una ontología es la adaptación de los términos de la ontología a las necesidades del proyecto. No es usual que al utilizar una ontología coincidan al cien por cien los conceptos, sus definiciones formales, las relaciones, los

atributos, los axiomas, etc. Los entornos software pueden ser de gran ayuda para realizar la adaptación, y el usuario debe estudiarlos con detenimiento, pues favorecerán o dificultarán la integración de la ontología en el proyecto. En gran parte, la posibilidad de utilizar estas herramientas software está supeditada, principalmente, al lenguaje en el que está implementada la ontología. Es decir, una ontología puede haber sido construida con la herramienta *A*, e implementada en el lenguaje *L*. Además, puede existir otra herramienta *B* capaz de importar la ontología implementada en el lenguaje *L*. Por lo tanto, el futuro usuario deberá estudiar las capacidades de ambas herramientas software.

Como ya se comentó en la sección 2.5.4, un **entorno de desarrollo de ontologías** puede contener varias herramientas software integradas y podrá incluir funcionalidades relacionadas con la creación, modificación o visualización de los términos, traductores para importar y exportar desde/a diferentes lenguajes, chequeo de la consistencia, programas para el control de acceso a usuarios, módulos software para el acceso a la ontología desde otras aplicaciones, etc.

En esta sección se proponen factores para valorar la idoneidad de los entornos software. Estos tratan: las prestaciones del entorno, aspectos de visualización de la ontología, de edición, de interacción con otras aplicaciones software, metodológicos, cooperativos, de traducción y de integración.

4.4.1.6.1. Características Descriptivas Relacionadas con la Dimensión "Entorno"

Además de las características que deben ser analizadas para determinar los factores anteriormente citados, existen otras características informativas y descriptivas:

- **Nombre del entorno**.
- **Compañías o entidades que lo han desarrollado**.
- **Descripción de las necesidades hardware y software** para utilizar el entorno, y el precio en adquirir este hardware y software.
- **Precio de las licencias de uso y de las interfaces de acceso.**
- **Fecha de salida al mercado**: fecha en la que se ha comercializado el entorno software.
- **Nombre de la metodología** en la que el entorno se basa para desarrollar ontologías.
- **Lenguajes clásicos y de marcado *web* desde los que importa, y a los que exporta el entorno.**
- **Lista de ontologías desarrolladas.**

4.4.1.6.2. Características de la Dimensión "Entornos" Usadas para la Medición de Idoneidad

4.4.1.6.2.1. Características que Influyen en el Factor "Prestaciones del Entorno"

En este factor se consideran las características que contribuyen a la calidad y fiabilidad de las aplicaciones software del entorno, y que harán que el proceso de adaptación requiera menos esfuerzo.

- **El entorno puede ser usado localmente**: la instalación exclusivamente en modo local supone un aumento en la seguridad de los datos almacenados y, en la mayoría de los casos, suelen ser aplicaciones más cómodas de manejar, y suelen usar más el modo gráfico. Sin embargo, tienen el inconveniente de que, si están instalados exclusivamente en modo local, no permiten la construcción cooperativa de ontologías.
- **El entorno puede ser usado en una red de área local**: la instalación en una red de área local permite a un equipo de trabajo desarrollar una ontología de forma cooperativa, aunque no permite que un equipo de usuarios trabajen en distintas localizaciones geográficas.
- **El entorno puede ser usado a través de Internet**: muchos proyectos software suelen desarrollarse por personas situadas en diferentes lugares geográficos. En algunos casos, la existencia de entornos, preparados para funcionar en Internet o en Intranet, que permitan trabajar con la ontología de forma cooperativa y desde varios lugares pueden resultar imprescindibles.
- **La facilidad en el uso del entorno**: el usuario debe tener en cuenta si al equipo de desarrollo del proyecto le va a costar mucho el aprendizaje del entorno. Y también, si las características de los entornos acelerarán o retrasarán el proceso de adaptación de la ontología. El usuario puede examinar, por ejemplo, la claridad de los comandos de las aplicaciones, la uniformidad de las opciones de los menús, la calidad de los manuales de ayuda, las ayudas contextuales, o los ejemplos de cómo usar las opciones.
- **El tiempo de respuesta del entorno**: el usuario debe analizar el tiempo que tardan los entornos en mostrar los términos o en actualizar las modificaciones. Esto suele tener más relevancia en entornos usados a través de Internet ya que, por ejemplo, entornos con gran cantidad de datos gráficos, o que permitan trabajar a un gran número de usuarios a la vez, pueden retrasar considerablemente los tiempos de respuesta.
- **La fiabilidad del entorno**. Un mal funcionamiento de los entornos software puede ocasionar un considerable retraso en el proceso de adaptación de la ontología a las necesidades del proyecto. Un entorno de reciente aparición, o que no haya sido

suficientemente probado, dará menos garantías que otro más utilizado. Para determinar esta característica, el usuario puede examinar: la fecha de la última versión del entorno, el número de ontologías desarrolladas usando el entorno, o los proyectos importantes en los que se ha elegido el entorno para construir o adaptar las ontologías.

4.4.1.6.2.2. Características que Influyen en el Factor "Aspectos de Visualización"

Este factor se refiere a la forma en la que el entorno muestra los términos, su contenido y la taxonomía de conceptos.

- **El entorno software muestra toda la información de todos los términos**: indica si el entorno muestra al usuario toda la información de todos los términos que se incluyen en la ontología.
- **El entorno software muestra los términos con el nivel de detalle requerido**: indica si el usuario puede seleccionar la información que quiere visualizar sobre los términos.
- **El entorno software muestra la taxonomía de conceptos**: se refiere a si el entorno permite visualizar la jerarquía de conceptos incluidos en la ontología.
- **El entorno software usa el modo gráfico para mostrar relaciones *ad-hoc* entre conceptos**: la visualización gráfica de relaciones *ad-hoc* entre conceptos y taxonomías de conceptos, ofrece una forma muy intuitiva de mostrar interconexiones entre conceptos. Normalmente, en este tipo de representación gráfica se muestran los conceptos como nodos de un grafo, y las relaciones entre conceptos como arcos del grafo.

4.4.1.6.2.3. Características que Influyen en el Factor "Aspectos de Edición"

Para realizar el proceso de adaptación se necesitarán añadir, modificar o eliminar términos de la ontología. En este factor se incluyen todas las facilidades que ofrecen los entornos para realizar estas tareas:

- **El entorno software utiliza todas las capacidades de representación del lenguaje**: no siempre los entornos consiguen aprovechar todas las posibilidades expresivas del lenguaje. Por ejemplo, en el lenguaje *RDF(S)* se puede dar un nombre alternativo a las clases con la etiqueta *label*; pero *OilEd 2.2*, a pesar de que puede exportar ontologías a este lenguaje, no utiliza esta etiqueta.
- **El entorno software permite hacer modificaciones en los términos en cualquier momento**: puede que algunos entornos no permitan cambiar los términos una vez que la ontología ha sido traducida a algún determinado lenguaje de implementación. Por ejemplo, la versión 1.0.3 de *OntoEdit* permite exportar ontologías a *FLogic*; pero, una vez realizada

esta acción, si un usuario manualmente o mediante otro entorno modifica la ontología en *FLogic*, *OntoEdit* 1.0.3 no puede acceder a la ontología modificada porque no dispone de la opción de importación desde ese lenguaje.

- **El entorno software permite realizar modificaciones en la taxonomía de conceptos de forma gráfica**: algunos entornos permiten definir la jerarquía entre conceptos mediante algún tipo de representación gráfica, normalmente mediante arcos representando relaciones taxonómicas (*subclase_de,* particiones exhaustivas y disjuntas). Esto ofrece al usuario una forma rápida e intuitiva de modificar la taxonomía de conceptos.
- **El entorno software permite realizar modificaciones en las relaciones *ad-hoc* de forma gráfica**: al igual que en el caso anterior, algunos entornos permiten hacer modificaciones de forma gráfica de las relaciones *ad-hoc* entre conceptos.

4.4.1.6.2.4. Características que Influyen en el Factor "Aspectos de Interacción"

En este factor se incluyen un conjunto de características que permiten a la aplicación software utilizar la ontología. Se han identificado dos posibilidades de uso: a través de ficheros o a través de interfaces de acceso (*APIs*). Las siguientes características permiten describir este factor:

- **La ontología se puede utilizar con independencia del entorno software**: se refiere a si la ontología se puede exportar y acceder a su contenido sin necesidad de utilizar ningún entorno software. La mayoría de los entornos que poseen esta característica suelen exportar la ontología a un fichero en formato *ASCII* o *XML*.
- **Se disponen de adecuadas interfaces de acceso a los términos**: se refiere a la existencia de módulos software que sirvan para acceder al contenido de la ontología, de forma local o remota.
- **Se dispone de información suficiente para utilizar las interfaces de acceso**: los módulos de acceso a las ontologías pueden tener manuales de referencia, ejemplos de uso de la interfaz, etc., que ayudarán al equipo de desarrollo a comprender su funcionamiento, para utilizarlo en su proyecto.
- **Las interfaces de acceso están disponibles**: se refiere a si los módulos de acceso se pueden conseguir sin dificultades, y si son código fuente abierto (*opensource*) para, llegado el caso, añadir o modificar las primitivas de acceso.
- **Se dispone de información suficiente para modificar las interfaces de acceso**: se refiere a la existencia de manuales adecuados, dirigidos al programador, que le permitirán con mayor facilidad modificar los módulos de acceso a los términos de la ontología.

4.4.1.6.2.5. Características que Influyen en el Factor "Aspectos Metodológicos"

Algunos entornos proporcionan soporte tecnológico a algunas actividades propuestas en las metodologías de desarrollo de ontologías. Las características de este factor muestran en qué medida los entornos ayudan a seguir la metodología de desarrollo. Y, más concretamente, el proceso de modificación de sus términos.

- **El entorno cubre todas las actividades del ciclo de vida de ontologías**: será más fácil para el usuario hacer modificaciones en la ontología (añadir, modificar o eliminar términos) si los entornos asisten en todas las actividades que indica la metodología de desarrollo.
- **El entorno cubre las actividades más importantes del ciclo de vida del desarrollo de ontologías**: algunas actividades, como la conceptualización, formalización e implementación, es muy recomendable realizarlas usando un entorno de desarrollo de ontologías apropiado, ya que pueden mostrar los términos existentes en cada momento, indicar al usuario las opciones disponibles y comprobar la consistencia una vez que los términos han sido modificados.
- **El entorno proporciona documentación suficiente al modificar los términos**: a la hora de realizar modificaciones en una ontología, es aconsejable disponer de informes sobre las modificaciones realizadas. Si los entornos crean estos documentos de forma automática, facilitarán esta actividad.
- **El entorno comprueba la consistencia al modificar los términos**: al realizar modificaciones en el modelo conceptual es muy importante la verificación de la consistencia en los cambios realizados.

4.4.1.6.2.6. Características que Influyen en el Factor "Aspectos Cooperativos"

El equipo de desarrollo del proyecto puede estar compuesto de varias personas, quizás situadas en diferentes lugares. Cuando una ontología es adaptada en estas circunstancias, es de gran ayuda que los entornos favorezcan el desarrollo cooperativo de la ontología. Se deben considerar las siguientes características en este factor:

- **El entorno permite crear grupos de trabajo para editar una misma ontología**: para modificar la ontología algunos entornos permiten definir grupos de trabajo, con los correspondientes derechos de acceso y modificación, informando qué integrantes del grupo están conectados, etc.
- **El entorno permite la edición simultánea de una ontología**: el proyecto puede requerir que el entorno permita incluir, modificar o borrar términos a la vez por varias personas del equipo. Por ejemplo, uno de los entornos que más favorece el trabajo colaborativo es

Tadzebao-WebOnto, que permite el uso de diagramas de discusión.

- **El entorno bloquea la escritura en la ontología para otros componentes del grupo cuando uno de ellos la está editando**: se refiere a si los entornos software bloquean totalmente el acceso de escritura cuando una persona del grupo de trabajo la está editando.
- **El entorno bloquea un término editado de la ontología para otros componentes del grupo cuando uno de ellos la está editando**: algunos entornos bloquean la escritura en el término concreto que está editando una persona del grupo de trabajo, permitiendo a otros miembros del grupo modificar otros términos.
- **El entorno software avisa de los cambios producidos en una ontología a todos los componentes del grupo**: en proyectos colaborativos es importante avisar al resto de los integrantes del grupo sobre las modificaciones realizadas utilizando, por ejemplo, mensajes de correo electrónico, una agenda común de cambios, etc.
- **El entorno gestiona el control de cambios**: de igual forma, es interesante que las herramientas registren quién ha realizado los cambios en cada uno de los términos, cuándo se han hecho tales cambios, así como la evolución de los cambios. De esta forma, se favorece el trabajo colaborativo.

4.4.1.6.2.7. Características que Influyen en el Factor "Aspectos de Traducción"

Si un usuario está buscando una ontología implementada en un lenguaje deberá considerar, no sólo el entorno en el que ha sido creada, sino otros entornos que puedan importar ese lenguaje. Además, puede encontrar una ontología desarrollada en un entorno que no tiene un lenguaje propietario; o quizás, desee desarrollar su aplicación accediendo a ontologías implementadas en lenguajes diferentes. En cualquier caso, deberá examinar las capacidades de traducción de los entornos de desarrollo. Cuando en este factor se habla de importación y exportación de lenguajes, se consideran lenguajes diferentes al lenguaje propietario del entorno. Algunos de los entornos existentes (como la versión comercial de *OntoEdit*, *Protégé2000* y *WebODE*) almacenan las ontologías en bases de datos, y facilitan alguna opción para exportar la ontología a uno o varios lenguajes.

- **El entorno importa ontologías implementadas en lenguajes "clásicos"**: puede darse el caso de que, a pesar de haber encontrado una ontología cuyo contenido coincida en gran parte con el contenido buscado, el entorno analizado no pueda importar desde el lenguaje clásico en el que está implementada.
- **El entorno importa ontologías implementadas en lenguajes de marcado *web***: al igual que en el caso anterior, se refiera a la capacidad del entorno para importar una ontología

representada en un lenguaje de marcado *web*.

- **El entorno exporta ontologías a lenguajes "clásicos" de implementación de ontologías**: una vez que se ha construido la ontología, puede ser que el proyecto requiera utilizar la ontología en un lenguaje clásico, diferente al propio del entorno software.
- **El entorno exporta ontologías a otros lenguajes de marcado *web***: de igual forma que en el caso anterior, puede ser que el proyecto requiera utilizar la ontología en un lenguaje de marcado *web*.
- **Los traductores minimizan la pérdida de conocimientos en la traducción**: si el traductor está bien construido, las únicas pérdidas de conocimientos serán debidas a una menor expresividad del lenguaje destino.
- **Los traductores permiten supervisar las traducciones**: normalmente las traducciones se realizan de forma automática; en esta característica se examina si el entorno pregunta al usuario cuando encuentra diferentes posibilidades de traducción, o avisa cuando detecta que se va a producir pérdida de conocimientos.

4.4.1.6.2.8. Características que Influyen en el Factor "Aspectos de Integración de Ontologías"

La mayoría de los entornos permiten integrar en una ontología, términos desde otras ontologías e incorporan algunas relaciones entre ellos [Pin00]. Otros posibilitan la mezcla (*merge*) de varias ontologías, de forma que se crea una ontología que unifica todos, o algunos, de los términos. En este caso, se construye una nueva ontología que une conceptos, relaciones, axiomas, etc., de otras ontologías existentes, sobre un mismo dominio [Pin00].

El proyecto software quizás necesite usar el conocimiento de varias ontologías, por lo que el equipo de desarrollo tendrá que realizar estos procesos de integración o mezcla. Se señalan aquí las características del entorno que deben examinarse relacionadas con esta cuestión.

- **Los entornos facilitan el proceso de integración**: se refiere a si el proceso de integración usando el entorno es fácil de realizar.
- **Los términos integrados en la ontología no necesitan ninguna notación especial**: se refiere a si, al realizar la integración por medio del entorno software, los términos incorporados desde otra ontología tienen una notación uniforme. Si no fuera así, el usuario deberá contemplar las distintas notaciones al acceder desde la aplicación a la ontología integrada.
- **El entorno permite seleccionar los términos que se desean integrar**: es posible que el usuario no desee integrar todos los términos de las ontologías, sino sólo una parte de ellos.

- **El entorno verifica la consistencia durante los procesos de integración o en los procesos de mezcla de ontologías**: si se realiza un proceso de integración, puede darse el caso de que los términos integrados, junto con las nuevas relaciones creadas en la integración, sean inconsistentes. En el proceso de mezcla, puede suceder que al considerar conjuntamente las definiciones de los términos de las ontologías que se van a mezclar, existan inconsistencias. Será de gran ayuda si el entorno detecta, entre otras, estas inconsistencias.

- **El entorno ofrece asistencia cuando se quiere hacer la mezcla "manual" de varias ontologías**: para favorecer el proceso de mezcla, sería conveniente que el entorno permitiera visualizar, preferiblemente de forma gráfica, la taxonomía de las ontologías que se desean mezclar y la ontología resultado de la mezcla, y que permitiera, fácilmente, modificar la nueva taxonomía formada.

- **El entorno permite hacer la mezcla automática de varias ontologías**: para favorecer el proceso de mezcla, sería conveniente que el entorno hiciera la mezcla semi-automática de los conceptos que tienen un cierto grado de semejanza. La única herramienta software encontrada hasta la fecha que realiza en parte este proceso es *Chimaera* [MGu00].

En la tabla 4.5 se muestra un resumen de todas las características expuestas en este apartado sobre la dimensión entorno software de desarrollo de ontologías.

Cód	Nombre de la característica	Tipo de valor	Cardinalidad	Autor	Refe-rencia
Características descriptivas					
T01	Nombre del entorno	Texto	(1,1)		
T02	Entidades desarrolladas	Texto	(0,1)		
T03	Requerimientos software necesarios	Texto	(0,1)		
T04	Requerimientos hardware necesarios	Fecha	(0,1)		
T05	Precio de requerimientos sw y hw ($)	Entero	(0,1)		
T06	Fecha de salida al mercado	Fecha	(0,1)		
T07	Precio de licencia de uso ($)	Entero	(0,1)		
T08	Precio de las interfaces de acceso ($)	Entero	(0,1)		
T09	Nombre de la metodología que asiste	Texto	(0,1)		
T010	Lista de lenguajes desde los que puede importar ontologías	Texto	(0,n)		
T011	Lista de lenguajes a los que puede exportar ontologías	Texto	(0,n)		
T012	Lista de ontologías desarrolladas	Texto	(0,n)		

Tabla 4.5. Características relacionadas con los entornos software de desarrollo de ontologías **(1/3)**.

Cód	Nombre de la característica	Tipo de valor	Cardinalidad	Autor	Refe-rencia
T1	**FACTOR: Prestaciones**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
T11	Instalación local	(No soportado, Soportado)	(1,1)	Duineveld et al.	[Dui99]
T12	Instalación en red local	(No soportado, Soportado)	(1,1)		
T13	Uso desde Internet	(No soportado, Soportado)	(1,1)		
T14	Facilidad de uso	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)	parcialmente por Duineveld et al. (*claridad de la interfaz, consistencia de la interfaz, claridad de los comandos, sistema de ayuda, ejemplos disponibles*)	[Dui99]
T15	Tiempo de respuesta	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	-parcialmente por Duineveld et al. (*velocidad de actualización*) -parcialmente Brown y Flett (*índices de búsqueda para mejorar la rapidez*)	[Dui99] [Bro01]
T16	Fiabilidad	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)	parcialmente por Duineveld et al. (*estabilidad*)	[Dui99]
T2	**FACTOR: Visualización**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
T21	Muestra la información de todos los términos	(No soportado, Soportado)	(1,1)	Duineveld et al.	[Dui99]
T22	Muestra el nivel de detalle requerido	(No soportado, Soportado)	(1,1)		
T23	Muestra la taxonomía	(No soportado, Soportado)	(1,1)	-Duineveld et al. (*visión general de la taxonomía*) -Brown y Flett (*visualización de la taxonomía*)	[Dui99] [Bro01]
T24	Muestra relaciones *ad-hoc*	(No soportado, Soportado)	(1,1)		
T3	**FACTOR: Edición**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
T31	Permite explotar todas las capacidades expresivas del lenguaje	(No soportado, Soportado)	(1,1)		
T32	Permite modificar la ontología en cualquier momento	(No soportado, Soportado)	(1,1)		
T33	Realiza modificaciones en la taxonomía de forma gráfica	(No soportado, Soportado)	(1,1)		
T34	Realiza modificaciones en relaciones ad-hoc de forma gráfica	(No soportado, Soportado)	(1,1)		
T4	**FACTOR: Interacción**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
T41	La ontología se puede utilizar con independencia del entorno	(No soportado, Soportado)	(1,1)		
T42	Adecuadas interfaces de acceso	(No soportado, Soportado)	(1,1)	-parcialmente por Duineveld et al. (*existencia de primitivas de alto nivel*) - parcialmente Brown y Flett (*escalabilidad*)	[Dui99] [Bro01]
T43	Información para usar las interfaces de acceso	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		
T44	Disponibilidad para modificar las interfaces de acceso	(No soportado, Soportado)	(1,1)		
T45	Información para modificar las interfaces de acceso	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		
T5	**FACTOR: Aspectos Metodológicos**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
T51	Cubre todas las actividades del ciclo de vida de la metodología	(No soportado, Soportado)	(1,1)		
T52	Cubre las actividades más importantes del ciclo de vida	(No soportado, Soportado)	(1,1)		
T53	Proporciona documentación suficiente al modificar los términos	(No soportado, Soportado)	(1,1)		
T54	Comprueba la consistencia de la ontología al realizar modificaciones	(No soportado, Soportado)	(1,1)	-Duineveld et al. -Brown y Flett	[Dui99] [Bro01]
T6	**FACTOR: Aspectos Cooperativos**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	Duineveld et al.	[Dui99]
T61	Permite crear grupos de trabajo	(No soportado, Soportado)	(1,1)		
T62	Permite edición simultánea de la ontología	(No soportado, Soportado)	(1,1)	-Duineveld et al. -Brown y Flett	[Dui99] [Bro01]
T63	Bloquea la escritura de la ontología en la edición	(No soportado, Soportado)	(1,1)	Duineveld et al.	[Dui99]
T64	Bloquea la escritura de los términos en la edición	(No soportado, Soportado)	(1,1)		
T65	Avisa de los cambios producidos	(No soportado, Soportado)	(1,1)		
T66	Gestión de cambios	(No soportado, Soportado)	(1,1)	Duineveld et al.	[Dui99]

Tabla 4.5. Características relacionadas con los entornos software de desarrollo de ontologías **(2/3)**.

Cód	Nombre de la característica	Tipo de valor	Cardinalidad	Autor	Refe-rencia
T7	**FACTOR: Traducción**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
T71	Importa desde lenguajes clásicos	(No soportado, Soportado)	(1,1)	parcialmente por Duineveld et al. (*importación desde otras herramientas*)	[Dui99]
T72	Importa desde lenguajes de marcado web	(No soportado, Soportado)	(1,1)	parcialmente por Duineveld et al. (*importación desde otras herramientas*)	[Dui99]
T73	Exporta a lenguajes clásicos	(No soportado, Soportado)	(1,1)	parcialmente por Duineveld et al. (*exporta a varios formatos*)	[Dui99]
T74	Exporta a lenguajes de marcado web	(No soportado, Soportado)	(1,1)	parcialmente por Duineveld et al. (*exporta a varios formatos*)	[Dui99]
T75	Minimiza pérdidas semánticas	(No soportado, Soportado)	(1,1)		
T76	Permiten traducciones supervisadas	(No soportado, Soportado)	(1,1)		
T8	**FACTOR: Integración**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
T81	Permite integración de ontologías	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)	parcialmente Brown y Flett (*adquisición de términos*)	[Bro01]
T82	Referencia de términos integrados	(MuyBaja, Baja, Media, Alta, MuyAlta)	(1,1)		
T83	Permite seleccionar los términos a integrar	(No soportado, Soportado)	(1,1)		
T84	Verifica la consistencia en la integración o en la mezcla	(No soportado, Soportado)	(1,1)		
T85	Asiste en la mezcla manual de ontologías	(No soportado, Soportado)	(1,1)		
T86	Asiste en la mezcla semi-automática de ontologías	(No soportado, Soportado)	(1,1)		

Tabla 4.5. Características relacionadas con los entornos software de desarrollo de ontologías **(3/3)**.

4.4.1.7. Características Relacionadas con la Dimensión "Costes de Uso de Ontologías"

En proyectos software, normalmente de índole comercial, el estudio de requisitos debe contemplar las estimaciones de gastos que va a ocasionar el desarrollo del proyecto. En los proyectos que usan ontologías, uno de los apartados que debería reflejar el estudio son las estimaciones de gastos relacionados específicamente con la elección de una determinada ontología. Las características que se recogen en esta dimensión se contabilizarán de forma aditiva, y el valor obtenido guardará una proporción inversa con la idoneidad en esta dimensión; es decir, independientemente del presupuesto con el que cuente el proyecto, a mayor coste de uso, menos idónea será la ontología en esta dimensión. Los factores que deben analizarse, mostrados en la tabla 4.6, son los siguientes:

- **Precio de adquisición de la ontología**: se refiere a los costes de las licencias de adquisición y explotación de la ontología.
- **Precio de los recursos necesarios para usar la ontología**: se refiere a las inversiones que deben efectuarse para adquirir recursos hardware (equipos especiales con diferentes capacidades de procesamiento, almacenamiento, comunicación, etc.) y software (diferentes sistemas operativos, programas de comunicaciones, etc.) específicos. También deberán incluirse en este apartado los gastos derivados de la contratación de personal especializado, cursos de formación, instalación, mantenimiento, etc.

- **Precio de adquisición de los módulos de acceso a la ontología**: se refiere a los costes de las licencias de adquisición, uso y modificación de las interfaces de acceso a los términos de la ontología.
- **Precio de adquisición del software especializado para adaptar la ontología**: va a depender de los procesos que sean necesarios realizar para adaptar la ontología al proyecto; por ejemplo, modificación del modelo conceptual, traducción a otros lenguajes de implementación, procesos de mezclas de ontologías, etc. En estos casos se debe calcular los costes de adquirir los entornos que ayuden a realizar estas tareas.

Cód	Nombre de la característica	Tipo de valor	Cardi-nalidad	Autor	Refe-rencia
Cost1	**FACTOR: Adquisición de la ontología**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
Cost2	**FACTOR: Recursos de uso**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
Cost3	**FACTOR: Adquisición de módulos de acceso**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		
Cost4	**FACTOR: Adquisición de software de adaptación**	(MuyBajo, Bajo, Medio, Alto, MuyAlto)	(1,1)		

Tabla 4.6. Características relacionadas con los costes de uso de ontologías.

4.4.2. Representación del Marco de Características en una Ontología: La *Reference Ontology*

Para que el usuario pueda completar las características usadas para la medición de idoneidad, indicadas en el marco de características, y para recoger la información de las características descriptivas, se ha construido una ontología en el dominio de las ontologías denominada ***Reference Ontology*** (RO). Existe una relación directa entre todas las características descriptivas identificadas en todas las dimensiones, y los atributos de instancia especificados en la RO. De igual forma, cada una de las características usadas para medir la idoneidad en las dimensiones "lenguajes", "entornos" y "metodologías", se corresponde con un atributo de instancia representado en la RO. En cambio, las características usadas para la medición de idoneidad identificadas para las dimensiones "contenido" y "costes" no tienen una correspondencia directa con los atributos de instancia de los conceptos almacenados en la RO; pero sí se puede valorar su idoneidad a partir de la información relacionada con el "contenido" de la RO, tal y como se describe en el apartado 4.7.1.3.2.

Para construir la RO se ha seguido la metodología de desarrollo de ontologías *METHONTOLOGY* ([Fer97] y [Gom98]), y se ha utilizado el entorno de desarrollo de ontologías *WebODE* [Arp01], que asiste en la aplicación de esta metodología. Los pasos que se han seguido para desarrollar la RO han sido los siguientes:

- **Especificación**: se determina en esta fase por qué se ha construido la RO y cuáles serán

sus usos y usuarios. La RO contendrá conocimientos sobre lenguajes usados en la implementación de ontologías, sobre metodologías y entornos software de desarrollo de ontologías, sobre varios aspectos de costes que pueden ocasionar el uso de las ontologías, y sobre el propio contenido de las ontologías existentes.

- **Conceptualización**: el objetivo de la conceptualización es estructurar los conocimientos del dominio, obteniendo un modelo conceptual que describe el problema y su solución. En esta fase, tomando como base el marco multinivel de características, se ha identificado el modelo conceptual de la RO. *METHONTOLOGY* propone que la conceptualización debe realizarse mediante un conjunto de representaciones intermedias (RRII) que formarán el modelo conceptual de la ontología [Fer00]. Las tablas que propone *METHONTOLOGY* para conceptuar son: "glosario de términos", "árbol de clasificación de conceptos", "diagrama de relaciones", "diccionario de conceptos", "tablas de relaciones", "tablas de atributos de instancia", "tablas de atributos de clase", "tablas de axiomas lógicos", "tabla de constantes", "tabla de fórmulas", "tabla de instancias" y "árboles de clasificación de atributos". El entorno *WebODE*, utilizado en el desarrollo de la RO, sirve para construir cada una de estas RRII, y realiza verificaciones de consistencia en cada RI y entre las distintas RRII. En esta sección se mostrarán las diferentes RRII obtenidas en el proceso de conceptualización de la RO.

- **Implementación**: el modelo conceptual (y las instancias) de la RO se encuentran almacenados en la base de datos del entorno software *WebODE*. Este entorno permite exportar e importar tanto el modelo conceptual como los conjuntos de instancias a varios lenguajes, entre ellos, un lenguaje de intercambio propio basado en XML, denominado *X-WebODE*. Las demás aplicaciones software que se han desarrollado aprovechan esta cualidad para usar la ontología RO, tal y como se comenta en la sección 4.3.

- **Evaluación**: *METHONTOLOGY* propone como tipos de evaluación: la verificación y la validación. La verificación consiste en determinar si el modelo conceptual de la RO, creado mediante las RRII, sigue unas reglas sintácticas dadas. Este proceso está asegurado debido a que *WebODE* realiza automáticamente la verificación de las RRII construidas en la conceptualización. La validación persigue determinar si el modelo conceptual de la RO carece de errores semánticos introducidos por el desarrollador de la ontología. La forma en la que se ha realizado este proceso es mediante la revisión, por varios desarrolladores de ontologías, de las características y su disposición en el marco multinivel de características. Sus opiniones han sido recogidas mediante un cuestionario, en el que los encuestados han identificado algunos errores conceptuales y carencias en el marco de características. El cuestionario se encuentra en el anexo I de este trabajo. No se ha realizado evaluación de la implementación porque los traductores de *WebODE* ya han sido

evaluados, y se sabe que el contenido generado es correcto.

En este apartado se muestran algunas tablas con las RRII que se han obtenido al realizar la conceptualización de la ontología RO. Se debe señalar que todos los nombres de términos, su descripción, propiedades, y demás elementos que aparecen en las tablas se encuentran en inglés. Esto es debido a que el objetivo de la RO es servir a los usuarios de ontologías como marco de referencia de búsqueda y elección de ontologías, y el idioma que predomina en este campo es el inglés.

En la RO están definidos, entre otros, los conceptos: *Language*, *Methodology*, *Tool*, cuyos atributos de instancia se corresponden con los criterios identificados en el marco multinivel de características sobre lenguajes, metodologías y entornos, respectivamente. El tipo de valor que pueden tomar estos atributos de instancia y su cardinalidad se mostraron en las tablas 4.2, 4.3, 4.4 y 4.5.

Además, en la RO están definidos los conceptos: *Ontology*, *Class*, *Attribute*, *Instance*, *Relation* y *Axiom*. Los atributos de instancia de estos conceptos no se corresponden (como para el caso de lenguajes, metodologías y entornos) con las características identificadas en el marco multinivel de características en la dimensión "contenido". En este caso, el usuario podrá determinar los valores en las características indicadas en el marco multinivel, a partir de los conocimientos incluidos en la RO. En la tabla 4.7 se expone, a modo de resumen, todos los atributos de instancia de los conceptos: *Ontology*, *Class*, *Attribute*, *Instance*, *Relation* y *Axiom* definidos en la RO, el tipo de valor que pueden tomar, y su cardinalidad.

Cód	Característica	Tipo de valor	Cardi-nalidad
	CONCEPTO: ONTOLOGY		
CO1	Name of the ontology	String	(1,1)
CO2	A short description of the ontology	String	(0,1)
CO3	The ontology is physically located at these servers	URL	(0,n)
CO4	Local URL for downloading the original ontology	URL	(0,n)
CO5	Status of the development of the ontology	{Study, In_Progress, Finished}	(1,1)
CO6	Type of ontology represented (van Heist classification)	{Application, Domain, Generic, Representation}	(1,n)
CO7	Domain: it describes the piece of reality that the builder of the ontology wants to represent.	String	(0,n)
CO8	Level of generality of the ontology	{Very_Low, Low, Medium, High, Very_High}	(0,1)
CO9	Purpose: It characterizes the different uses of the ontology.	String	(0,n)
CO10	Formality: It expresses the level of formality of the ontology (Uschold scale).	String	(0,1)
CO11	List of ontologies integrated in the ontology	String	(0,n)
CO12	Kind of formalism paradigms used to formalize the ontology	String	(0,n)
CO13	Companies or entities which have developed the ontology	String	(0,n)
CO14	Knowledge sources used in the knowledge acquisition process	String	(0,n)
CO15	Strategy used to identifying the terms	{Bottom-up, Top-down, Middle-out}	(0,1)
CO16	Announced release date of the developed ontology	Date	(0,1)
CO17	Date of the last modification of the ontology (changing terms)	Date	(0,1)
CO18	List of important projects which use the ontology	String	(0,n)
CO19	The number of concepts (classes) represented in the ontology	Integer	(1,1)
CO20	The number of instances of concepts represented in the ontology	Integer	(1,1)
CO21	Average number of attributes in concepts	Integer	(1,1)
CO22	The number of relations defined in the ontology	Integer	(1,1)
CO23	The number of functions defined in the ontology	Integer	(1,1)
CO24	The concepts of the ontology are classified from several perspectives.	{Single, Multiple}	(1,1)
CO25	Number of instances of the relation Subclass-of (or another similar one)	Integer	(1,1)
CO26	Number of instances of the relation Not-Subclass-of (or another similar one)	Integer	(1,1)
CO27	Number of exhaustive partition is used in the ontology	Integer	(1,1)
CO28	Number of disjoint partition is used in the ontology	Integer	(1,1)
CO29	The number of axioms defined as independent elements (those not linked to concepts).	Integer	(1,1)
CO30	There are defined axioms to solve queries	Boolean	(0,1)
CO31	There are defined axioms to infer knowledge about attributes of concepts.	Boolean	(0,1)
CO32	There are defined axioms used to verify the consistency of terms in the ontology	Boolean	(0,1)
CO33	Price of the licence for ontology use (in US $)	Integer	(0,1)
	CONCEPTO: CLASS		
CC1	Name of the class	String	(1,1)
CC2	Description of the class	String	(0,1)
CC3	Formal description of the class (literal specification)	String	(0,1)
	CONCEPTO: ATTRIBUTE		
CA1	Name of the attribute	String	(1,1)
CA2	Description of the attribute	String	(0,1)
CA3	Type of attribute	{Instance, Class, Local, Global, Polymorphic}	(0,n)
CA4	Formal description of the attribute (literal specification)	String	(0,1)
	CONCEPTO: INSTANCE		
CI1	Name of the instance	String	(1,1)
CI2	Description of the instance	String	(0,1)
	CONCEPTO: RELATION		
CR1	Name of the relation	String	(1,1)
CR2	Description of the relation	String	(0,1)
CR3	It is a function	Boolean	(1,1)
CR4	Relation arity	Integer	(0,1)
CR5	Formal properties specified	{Reflexive, Irreflexive, Symmetrical, Asymmetrical, Antisymmetrical, Transitive, Intransitive }	(0,n)
CR6	Formal description of the relation (literal specification)	String	(0,1)
	CONCEPTO: AXIOM		
CNA1	Name of the named-axiom	String	(1,1)
CNA2	Description of the named-axiom	String	(0,1)
CNA3	Formal description of the named-axiom (literal specification)	String	(0,1)

Tabla 4.7. Atributos de instancia de los conceptos relacionados con el "contenido" de la RO, junto con el tipo de valor y cardinalidad.

A continuación se muestran las siguientes RRII: en la tabla 4.8 la "ficha de descripción general" de la ontología RO; en la tabla 4.9 el "glosario de términos" con algunos términos a modo de ejemplo; en la tabla 4.10 se expone parte del "diccionario de conceptos"; en las figuras 4.4 y 4.5, las relaciones directas e inversas entre los conceptos del dominio; en las tablas 4.11, 4.12 y 4.13 se muestran algunas relaciones binarias en las "tablas de relaciones"; en las tablas 4.14, 4.15, y 4.16 se exponen tres ejemplos de "tablas de atributos de instancia"; y en la tabla 4.17 se muestran como ejemplo algunas instancias en la "tabla de instancias".

En la conceptualización de la RO no se han usado las siguientes RRII: el "árbol de clasificación de conceptos" no se muestra debido a que no existen conceptos subclases de otros conceptos; no se ha especificado ningún atributo de clase, por lo que no se muestran las "tablas de atributos de clase"; tampoco se han identificado axiomas, constantes, ni fórmulas, por lo que no se muestran las "tablas de axiomas", las "tablas de constantes", ni las "tablas de fórmulas"; y tampoco se da el caso de deducir atributos a partir de otros atributos, por lo que no aparecen "árboles de clasificación de atributos".

Nombre	Reference Ontology
Fecha de creación	September 2001
Autores	Adolfo Lozano Tello and the Knowledge Reuse Group (AI Lab) from the Technical University of Madrid
Descripción	It is a domain ontology about ontologies. Its purpose is to contain knowledge about different dimensions of properties to select ontologies. It gets a living set of features together to allow us characterize ontologies from the user point of view and have a common logical organization.

Tabla 4.8. Ficha de descripción general de la ontología RO.

Nombre	Descripción
Ontology	Ontologies are defined as "a formal, explicit specification of a shared conceptualization" [Stu98], that is, Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.
Class	It is also known as concept or category, is used in a broad sense. They can be abstract or concrete, elementary or composite, real or fictitious. A concept can be anything about which something is said, and, therefore, could also be the description of a task, function, action, strategy, reasoning process, etc.
Attribute	It is a property of a concept.
Instance	It is a term used to represent elements in a domain.
Axiom	It is a model sentence that is always true. They are included in an ontology for several purposes, such as constraining its information, verifying its correctness or deducting new information.
Relation	Represent a type of association between concepts of the domain. They are formally defined as any subset of a product of n sets.
Language	It is the formal language used to specify ontologies.
Methodology	It is a methodology used to develop ontologies.
Tool	It is a software program (or software programs) used to develop ontologies.
Ontology Included	It is a relation that links an ontology with the ontologies included in.
Ontology Developed With Methodology	It is a relation that links an ontology with the methodology used to develop it.
Tool Exports to Languages	It is a relation that links a tool with the languages, which can export the ontologies.
Type Of Ontology	Type of ontology represented (van Heist classification): Application, Domain, Generic, Representation.
Number Of Relations	The number of relations defined in the ontology
Strategy For Building Ontologies	Strategy used in the methodology to identifying the terms: Bottom-up, Top-down, Middle-out.
Domain	Domain describes the piece of reality that the builder of the ontology wants to represent.
Developers	Companies or entities, which have developed the ontology.
URL Location	URL for downloading or visualizing the ontology.
Taxonomy	They are widely used to organize ontological knowledge in the domain using generalization/specialization relationships through which simple/multiple inheritance could be applied.
SHOE	It stands for Simple HTML Ontology Extension. It was developed first as an extension of HTML, with the aim of incorporating machine-readable semantic knowledge in HTML or other www documents. Recently, it has been adapted in order to be XML compliant. The intent of this language is to make possible for agents to gather meaningful information about web pages and documents. [Luk00].
Uschold-King methodology	This methodology provides guidelines to build ontologies. It proposes to identify the purpose of the ontology, building, evaluating and documentation.
Bibliographic_data	It is an ontology in the domain of publications, represented in DAML+OIL. It is a Simple ontology for describing publications, loosely based on the *bibtex* definitions in the Latex book.
•••	•••

Tabla 4.9. Parte del glosario de términos de la ontología RO.

Nombre del concepto	Sinónimo	Abreviaturas	Instancias	Atributos de clase	Atributos de instancia	Relaciones
Ontology			-Abstract-Algebra -Agents -Bibliographic-Data -Chemical-Elements -Components-Assemblies -Device-Ontology …		-Domain -Developers -Licence Price -Purpose -URL Location -Type of ontology …	-Ontologies Included -Is Included In Ontologies -Ontology Developed With Tools …
Class	-Concept -Category		-Abelian-Group -Organization -Book -Author -Metals -Structural-Element -Vehicles …		-Class Name -Class Description	-Class Is Subclass Of -Class Is Superclass Of -Class Belongs To Ontology - Class Has Attributes …
Attribute			-Description of Group -Jobs -Pages of the book -Atomic-Weight -Publication-Date …		-Attribute Name -Attribute Description	-Is Attribute Of Class -Attribute Belongs To Ontology
Instance	Individual		-Signal-Shape -Mercury -Diode -Ford …		-Instance Name -Instance Description	-Is Instance Of Class -Instance Belongs To Ontology
Axiom			-Group-Alk -Named-Axiom-Frame -Car-after-1964-Require-Smog-Checks-Axiom …		-Named-axiom Name -Named-axiom description	-Named-axiom Links to Concept -Named-axiom Belongs To Ontology
Relation			-Has-Name -Has-Author -Antisymmetric -Boiling-Point -Has Components …		-Relation Name -Relation Description -Is A Function -Relation Arity …	-Relation Belongs to Ontology -Relates Classes
Language			-Ontolingua -LOOM -FLogic -SHOE -XOL -DAML+OIL …		-Language Name -Type Of Language -Has Metaclasses …	-Language Represents Ontologies -Languages Is Exportable From Tools …
Methodology			-CYC Methodology -METHONTOLOGY -Uschold-Grüninger -Bernaras et al -SENSUS-based …		-Methodology Name -Recommended Techniques …	-Methodology Is Used to Develop Ontologies - Methodology Assisted By Tools
Tool			-Ontolingua Server -OilEd -OntoEdit -WebODE …		-Tool Name -Date Of Release -Local Use …	-Tool Exports to Languages -Tool Used to Develop Ontologies …

Tabla 4.10. Parte del diccionario de conceptos de la ontología RO.

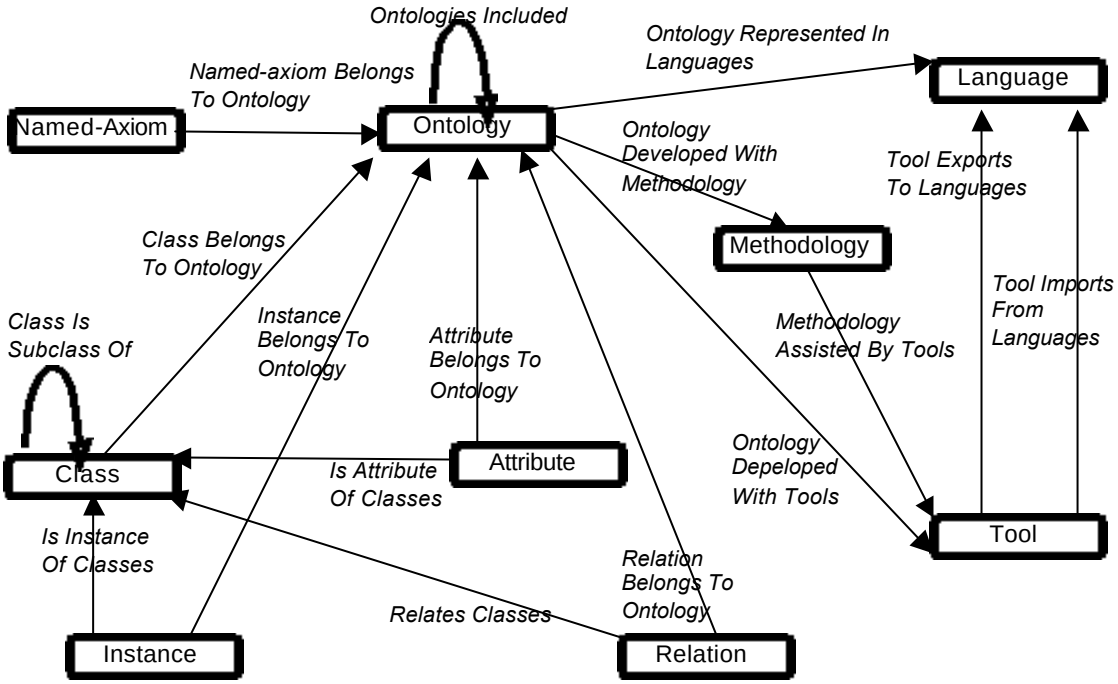


Figura 4.4. Diagrama de relaciones directas entre conceptos en la ontología RO.

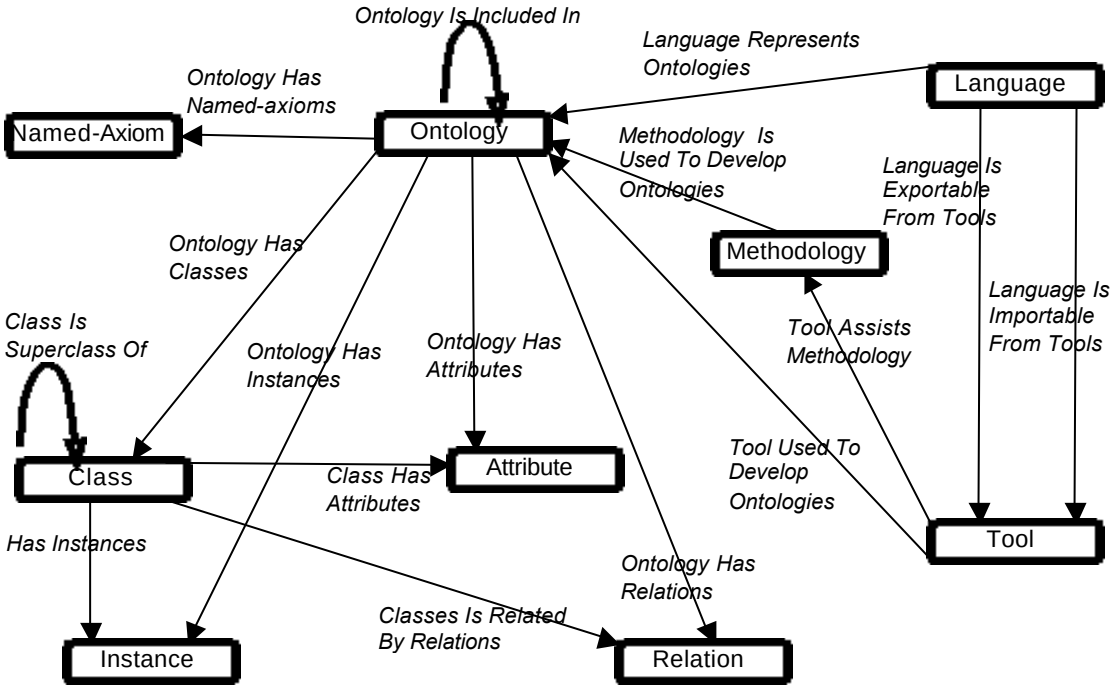


Figura 4.5. Diagrama de relaciones inversas entre conceptos en la ontología RO.

Nombre de la relación	Ontologies Included
Concepto origen	Ontology
Cardinalidad del origen	(0,n)
Concepto destino	Ontology
Propiedades matemáticas	--
Relación inversa	Ontology Is Included In
Referencias	--

Tabla 4.11. Relación binaria *Ontologies Included* en la ontología RO.

Nombre de la relación	Ontology Developed With Methodology
Concepto origen	Ontology
Cardinalidad del origen	(0,1)
Concepto destino	Methodology
Propiedades matemáticas	--
Relación inversa	Methodology Is Used To Develop Ontologies
Referencias	--

Tabla 4.12. Relación binaria *Ontology Developed With Methodology* en la ontología RO.

Nombre de la relación	Tool Exports To Languages
Concepto origen	Tool
Cardinalidad del origen	(0,n)
Concepto destino	Language
Propiedades matemáticas	--
Relación inversa	Language Is Exportable From Tools
Referencias	--

Tabla 4.13. Relación binaria *Tool Exports To Languages* en la ontología RO.

Nombre del atributo de instancia	Type Of Ontology
Concepto	Ontology
Tipo de valor	{Application, Domain, Generic, Representation}
Unidad de medida	--
Precisión	--
Intervalo de valores	--
Valor por defecto	--
Cardinalidad	(1,n)
Deducido de los atributos de instancia	--
Deducido de los atributos de clase	--
Deducido de las constantes	--
Fórmula	--
Para deducir	--
Referencias	[vHe97] van Heist G., Schreiber A., y Wielinga B., "Using Explicit Ontologies in KBS Development", International Journal Human-Computer Studies, vol 45, 183-292, 1997.

Tabla 4.14. Tabla del atributo de instancia *Type of Ontology* en la ontología RO.

Nombre del atributo de instancia	Number Of Relations
Concepto	Ontology
Tipo de valor	Integer
Unidad de medida	--
Precisión	--
Intervalo de valores	--
Valor por defecto	--
Cardinalidad	(1,1)
Deducido de los atributos de instancia	--
Deducido de los atributos de clase	--
Deducido de las constantes	--
Fórmula	--
Para deducir	--
Referencias	--

Tabla 4.15. Tabla del atributo de instancia *Number of Relations* en la ontología RO.

Nombre del atributo de instancia	Strategy For Building Ontologies
Concepto	Methodology
Tipo de valor	{Application-dependent, Application-semidependent, Application-independent}
Unidad de medida	--
Precisión	--
Intervalo de valores	--
Valor por defecto	--
Cardinalidad	(0,1)
Deducido de los atributos de instancia	--
Deducido de los atributos de clase	--
Deducido de las constantes	--
Fórmula	--
Para deducir	--
Referencias	[Fer99b] Fernández López M., "Overview of Methodologies for Building Ontologies", Proc. of the IJCAI'99 Workshop on Ontologies and PSMs, 4-1:4-13, Stockholm, Sweden, Ago, 1999.

Tabla 4.16. Tabla del atributo de instancia *Strategy For Building Ontologies* en la ontología RO.

Instancia	Atributo	Relación	Valor
Bibliographic-Data (Ontology)	Type Of Ontology	--	Domain
	Number Of Relations	--	16
	Type Of Taxonomy	--	Single
	--	Ontologies Included	Simple-Time Agents Frame-Ontology Documents
	…	…	…
Assignment-Object (Ontology Class)	Class Name	--	Assignment-Object
	Class Description	--	ASSIGNMENT-OBJECT denotes an object appearing generally in assignment problems.
	--	Class Belongs To Ontology	Job-Assignment-Task
	…	…	…
Title-Of (Attribute Class)	Attribute Name	--	Title Of
	--	Attribute Belongs To Ontology	Bibliographic Data
	--	Is Attribute Of Class	Article Reference
	…	…	…
Mercury (Ontology Instance)	Instance Name	--	Mercury
	Instance Description	--	… Mercury is the only common metal liquid at ordinary temperatures. …
	--	Is Instance Of Classes	Third-Transition-Series
	…	…	…
Ontolingua (Ontology Language)	Representation Formalism	--	Frame-based Classical Logic
	Has Metaclasses	--	Supported
	Can Define Global Attributes	--	Supported
	--	Language Is Exportable From Tools	Ontolingua Editor
	…	…	…
Bernaras et al Methodology (Ontology Methodology)	Steps To Build Ontologies	--	Specification of the application | Preliminary design based in top-level categories | Ontology refinement and structuring
	Strategy For Building Ontologies	--	Application-dependent
	--	Methodology Assisted By Tools	--
	…	…	…
OilEd (Ontology Tool)	Date Of Release	--	2-7-2001
	Internet-based Use	--	Non-Supported
	Tool Shows Taxonomy Graphically	--	Supported
	--	Tool Imports From Languages	-DAML -RDFS
	…	…	…
…	…	…	…

Tabla 4.17. Tabla de instancias con algunos ejemplos de la ontología RO.

Las tablas completas con instancias de "lenguajes", "metodologías" y "entornos software" se muestran en el apartado 4.5.2. Todas las características indicadas en el marco multinivel de características de estas tres dimensiones se han implementado como atributos de instancia en la RO. Si embargo, las características de las dimensiones "contenido" y "costes" no se han implementado tal y como aparecen en el marco multinivel de características. Esto se debe a que en el marco multinivel de características se incluyen características que el usuario valorará según las necesidades de su proyecto. La RO almacena la información sobre los conceptos: *Ontology*, *Class*, *Attribute*, *Instance*, *Relation* y *Axiom* que va a permitir al usuario completar los valores de las características relacionadas con las dimensiones "contenido" y "costes"; pero, debido a su gran cantidad de instancias, no se incluyen en esta memoria.

4.5. INSTANCIACIÓN DE LA *REFERENCE ONTOLOGY*

La RO ha sido instanciada para las dimensiones de contenido, lenguajes, metodologías, entornos de desarrollo y costes de uso. De esta forma, el usuario que necesite ontologías no deberá realizar el esfuerzo de buscar esta información, heterogénea y dispersa, en distintos sitios *web* y servidores de ontologías. Las instancias de la RO que se muestran en esta sección deberán ser actualizadas a medida que aparezcan nuevos elementos o cambien las características de los ya existentes. El proceso de instanciación se ha realizado de dos formas:

a) Para la dimensión "contenido" se han recopilado las ontologías existentes y disponibles en los servidores *web* de ontologías, las cuales se encuentran implementadas en diferentes lenguajes. Se han desarrollado ***wrappers*** que extraen automáticamente la información relevante de estas ontologías y las transforman en instancias acordes con el modelo conceptual de la RO. Estas instancias se representan en el formato *X-WebODE*, que pueden ser importadas desde el entorno *WebODE*. Se han desarrollado *wrappers* (a partir de ahora se referenciarán como ***Ontowrappers***) para los lenguajes de implementación: *Ontolingua*, *DAML* y el propio *X-WebODE*, por ser estos los lenguajes en los que están implementadas las ontologías encontradas en: *Ontolingua Server* (*Ontolingua*), en el servidor *WebODE* (*X-WebODE*), en la página *web* de *SHOE* (*DAML*), en la del *Ontology Group at ITBM-CNR* (*Ontolingua*) y en la de *DAML Ontology Library* (*DAML*).

   Usando los *Ontowrappers*, se ha extraído la información sobre los conceptos, atributos, instancias, relaciones, taxonomía y axiomas representados en la ontología, su descripción, disposición, y propiedades, y cómo están relacionados entre ellos. Además, se han completado automáticamente algunos atributos de instancia de las ontologías. Otras características de las ontologías han sido completadas manualmente debido a que, o bien no

se puede representar en alguno de los lenguajes, o porque esta información no está definida explícitamente en la implementación de la ontología. En la tabla 4.18 se muestran las características de los conceptos extraídas por los *Ontowrappers* de forma automática y se indican las que han sido completadas de forma manual. En la columna *Relleno* se indica: "A" para indicar que la característica ha sido completada automáticamente por los *Ontowrappers*, "M" si siempre ha sido completada manualmente usando *WebODE*, o "D" cuando se ha completado manualmente sólo en algunos casos, debido a que algunas ontologías o lenguajes no disponían, a veces, de esa información. En la tabla 4.19 se muestran las relaciones entre los conceptos que se han instanciado en la RO, también de forma automática, usando los *Ontowrappers*.

Cód	Relleno	Característica
		CONCEPTO: ONTOLOGY
CO1	A	Name of the ontology
CO2	A	A short description of the ontology
CO3	A	The ontology is physically located at these servers
CO4	A	Local URL for downloading the original ontology
CO5	D	Status of the development of the ontology
CO6	M	Type of ontology represented (van Heist classification)
CO7	M	Domain: it describes the piece of reality that the builder of the ontology wants to represent.
CO8	D	Level of generality of the ontology
CO9	M	Purpose: It characterizes the different uses of the ontology.
CO10	D	Formality: It expresses the level of formality of the ontology (Uschold scale).
CO11	A	List of ontologies integrated in the ontology
CO12	A	Kind of formalism paradigms used to formalize the ontology
CO13	D	Companies or entities which have developed the ontology
CO14	M	Knowledge sources used in the knowledge acquisition process
CO15	M	Strategy used to identifying the terms
CO16	D	Announced release date of the developed ontology
CO17	D	Date of the last modification of the ontology (changing terms)
CO18	M	List of important projects which use the ontology
CO19	A	The number of concepts (classes) represented in the ontology
CO20	A	The number of instances of concepts represented in the ontology
CO21	A	Average number of attributes in concepts
CO22	A	The number of relations defined in the ontology
CO23	A	The number of functions defined in the ontology
CO24	A	The concepts of the ontology are classified from several perspectives.
CO25	A	Number of instances of the relation Subclass-of (or another similar one)
CO26	A	Number of instances of the relation Not-Subclass-of (or another similar one)
CO27	A	Number of exhaustive partition is used in the ontology
CO28	A	Number of disjoint partition is used in the ontology
CO29	A	The number of axioms defined as independent elements (those not linked to concepts).
CO30	M	There are defined axioms to solve queries
CO31	M	There are defined axioms to infer knowledge about attributes of concepts.
CO32	M	There are defined axioms used to verify the consistency of terms in the ontology
CO33	M	Price of the licence for ontology use (in US $)
		CONCEPTO: CLASS
CC1	A	Name of the class
CC2	A	Short description of the class
CC3	A	Formal description of the class,(literal specification
		CONCEPTO: ATTRIBUTE
CA1	A	Name of the attribute
CA2	A	Short description of the attribute
CA3	A	Type of attribute
CA4	A	Formal description of the attribute (literal specification)
		CONCEPTO: INSTANCE
CI1	A	Name of the instance
CI2	A	Short description of the instance

Tabla 4.18. Características sobre los conceptos de la RO que son completadas manualmente o automáticamente mediante los *OntoWrappers* **(1/2)**.

Cód	Relleno	Característica
		CONCEPTO: RELATION
CR1	A	Name of the relation
CR2	A	Short description of the relation
CR3	A	It is a function
CR4	A	Relation arity
CR5	A	Formal properties specified
CR6	A	Formal description of the relation (literal specification)
		CONCEPTO: AXIOM
CNA1	A	Name of the named-axiom
CNA2	A	Short description of the named-axiom
CNA3	A	Formal description of the named-axiom (literal specification)

Tabla 4.18. Características sobre los conceptos de la RO que son completadas manualmente o automáticamente mediante los *OntoWrappers* **(2/2)**.

Cód	Relación	Concepto origen	Concepto destino	Cardina-lidad	Relación inversa
R1	OntologiesIncluded	Ontology	Ontology	(0,n)	OntologyIsIncludedIn
R2	ClassBelongsToOntology	Class	Ontology	(1,1)	OntologyHasClasses
R3	ClassIsSubclassOf	Class	Class	(0,n)	ClassIsSuperclassOf
R4	AttrubuteBelongsToOntology	Attribute	Ontology	(1,1)	OntologyHasAttributes
R5	IsAttributeOfClass	Attribute	Class	(0,n)	ClassHasAttributes
R6	InstanceBelongsToOntology	Instance	Ontology	(1,1)	OntologyHasInstances
R7	IsInstanceOfClasses	Instance	Class	(0,n)	HasInstances
R8	RelationBelongsToOntology	Relation	Ontology	(1,1)	OntologyHasRelations
R9	RelatesClasses	Relation	Class	(1,n)	ClassesAreRelatedByRelations
R10	Named-axiomBelongsToOntology	Axiom	Ontology	(1,1)	OntologyHasNamed-axioms
R11	OntologyRepresentedInLanguages	Ontology	Language	(0,n)	LanguageRepresentsOntologies
R12	OntologyDevelopedWithMethodology	Ontology	Methodology	(0,1)	MethodologyIsUsedToDevelopOntologies
R13	OntologyDevelopedWithTools	Ontology	Tool	(0,n)	ToolUsedToDevelopOntologies

Tabla 4.19. Características sobre las relaciones entre conceptos de la RO que son completadas en la dimensión "contenido" automáticamente con los *OntoWrappers*.

b) Por otro lado, para las dimensiones de lenguajes de implementación, metodologías de desarrollo, entornos software de desarrollo de ontologías y costes de uso se ha analizado la bibliografía existente, se han examinado directamente estos elementos, y se ha consultado a expertos en estos campos. Usando el entorno *WebODE* se han completado manualmente los valores que toman cada una de las instancias identificadas.

El modelo conceptual y las instancias identificadas en la RO servirán para que el usuario pueda buscar, a partir de los objetivos fijados para su proyecto, las ontologías más apropiadas. Como se expondrá en el apartado 4.6.1, se ha desarrollado una aplicación software para ayudar a realizar esta tarea.

4.5.1. Instanciación de la Dimensión "Contenido" de Ontologías

Como ya se ha comentado, el objetivo de este proceso consiste en extraer la información relevante de las ontologías existentes situadas en servidores de ontologías, codificadas en un

determinado lenguaje, y modelizar dicho contenido como instancias de la dimensión "contenido" de la RO. Para efectuar este proceso se han realizado las siguientes tareas:

- Seleccionar los servidores y *sites* en los se encuentran almacenadas las ontologías.
- Recopilar las ontologías existentes y disponibles en los distintos servidores de ontologías seleccionados. Estas ontologías se han exportado desde los servidores y se han almacenado localmente en ficheros.
- Identificar, para cada lenguaje, los patrones sintácticos que permiten transformar la representación de una ontología, en instancias válidas del modelo conceptual de la dimensión "contenido" de la RO.
- Se han implementado un conjunto de *wrappers* que extraen automáticamente la información relevante de las ontologías, usando los patrones sintácticos previamente identificados. Dichos *wrappers* convierten esta información en instancias válidas de la dimensión "contenido" de la RO. A estos programas se les ha denominado ***OntoWrappers***.
- La información producida por los *OntoWrappers* se ha importado en el servidor *WebODE* como instancias de la dimensión "contenido" de la RO. *WebODE* permite importar conjuntos de instancias, de forma que las instancias de esta dimensión pueden ser incorporadas a *WebODE*, a medida que cada *OntoWrapper* genera los ficheros de importación.

Fuentes de conocimientos para la dimensión "contenido" de la RO

Tras analizar las direcciones *web* sobre ontologías y servidores de ontologías existentes, se encontraron librerías de ontologías accesibles únicamente en: el servidor de ontologías *Ontolingua Server*, en el servidor *WebODE*, en las página *web* de *SHOE*, del *Ontology Group at ITBM-CNR* y de *DAML Ontology Library*. Los lenguajes en los que están implementadas las ontologías encontradas fueron:

- ***Ontolingua***: Se han encontrado 115 ontologías en el *Ontolingua Server*, y 41 ontologías accesibles en la página del *Ontology Group at ITBM-CNR* en el proyecto *ON9*.
- ***DAML***: Se han encontrado 120 ontologías catalogadas en la *DAML Ontology Library* y 8 ontologías en la versión de *DAML* en la página *web* de *SHOE.*
- ***X-WebODE***: Se han encontrado 15 ontologías en el servidor de ontologías *WebODE*.

Para realizar la extracción automática de la información relevante de las ontologías representada en estos lenguajes se han desarrollado las aplicaciones software denominadas:

Ontolingua_OntoWrapper, ***DAML_OntoWrapper*** y ***X-WebODE_OntoWrapper***, para los lenguajes *Ontolingua*, *DAML* y *X-WebODE*, respectivamente. De esta forma, se consiguen las instancias de la dimensión "contenido" de ontologías conforme al modelo conceptual especificado en la RO, tal como se representa en la figura 4.6. La descripción del funcionamiento de los *OntoWrappers* se presenta en el anexo II de este trabajo.

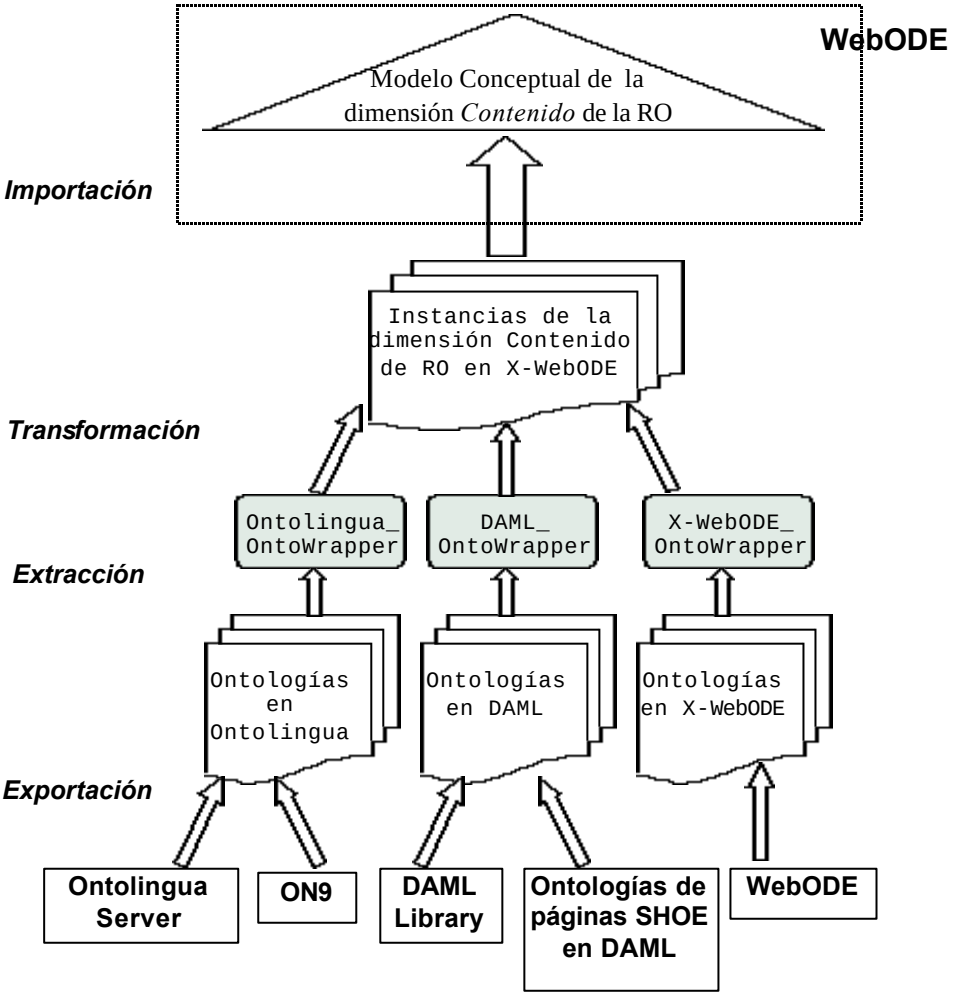


Figura 4.6. Esquema del proceso de instanciación de la dimensión "contenido" de la RO.

4.5.2. Instanciación de las Dimensiones: "Lenguaje", "Metodología", "Entorno "y "Costes"

Las dimensiones de lenguajes, metodologías, entornos de desarrollo y costes de uso de ontologías han sido instanciadas directamente utilizando el entorno *WebODE*.

Para las instancias de lenguajes se ha utilizado la información disponible en [Cor00], e información procedente de los lenguajes: *Ontolingua* [Far96], *CycL* [Len90], *LOOM* [MGr91], *Flogic* [Kif95], *OCML* [Mot95], *OKBC* [Cha97], *XOL* [Kar99], *SHOE* [Luk00], *OIL* [Hor00],

OML/CKML [Ken98], *DAML+OIL* [Hor01] y *RDF(S)* [Las99]. Los valores de las características descriptivas y las relacionadas con el factor "conocimiento del dominio" de los lenguajes previamente citados se muestran en las tablas 4.20 y 4.21, para los lenguajes clásicos y de marcado *web*, respectivamente; y las características relacionadas con el factor "mecanismos de inferencia" se muestran en las tablas 4.22 y 4.23.

Para introducir la información sobre metodologías existentes, se ha seguido el estudio de [Fer99a], y publicaciones sobre las metodologías *CYC* [Len90], Uschold y King [Usc95], Grüninger y Fox [Gru95b], la propuesta de Bernaras y colegas [Ber96], la metodología de *SENSUS* [Swa97], *METHONTOLOGY* [Fer97] y [Gom98], *Co4 Methodology* [Euz95] y *(KA)*^{2} *Methodology* [Ben99]. Los valores de las características que toman las instancias de metodologías se muestran en la tabla 4.24 (características relacionadas con la precisión, usabilidad y madurez de la metodología) y 4.25 (características descriptivas).

La información sobre entornos de desarrollo de ontologías se ha extraído del estudio presentado en [Dui99], del *workshop* de la red temática *OntoWeb*: "Universal Ontology Editor Workshop", y analizando las publicaciones sobre los siguientes entornos: *Ontolingua Server* [Far96], *Tadzebao-WebOnto* [Dom98], *OntoEdit* [Sta00], *WebODE* [Arp01], *OILed* [Bec01], y *Protégé2000* [Noy01], y probando directamente estas herramientas. Los valores de las características, para varias de las instancias citadas, pueden verse en la tabla 4.26.

La información sobre las características relacionadas con los costes de uso se han obtenido buscando en las páginas *web* correspondientes sobre las ontologías y entornos de desarrollo de ontologías. Los valores de estas instancias se encuentran en las características descriptivas de las instancias de "contenido" (CO33) para cada ontología, y en las instancias de la dimensión "entornos". Las características sobre costes en las instancias de "entornos" aparecen en la tabla 4.26 con los códigos T05, T07 y T08.

Cód	Característica	Ontolingua	OKBC	OCML	LOOM	FLogic
L0	**DESCRIPTIVE CHARACTERISTICS**					
L01	Language_Name	Ontolingua	OKBC	OCML	LOOM	Flogic
L02	Type_Of_Language	Classical	Classical	Classical	Classical	Classical
L03	Representation_Formalisms	Frame-based, Classical_logic	Frame-based	Frame-based, Classical_logic	Description _Logic	Frame-based, Classical_logic
L04	Help-Manuals	High	High	Medium	Medium	Medium
L05	Allows_Use_Independently_From_Its_Environment	Supported	Supported	Supported	Supported	Supported
L06	Language_Information_URL	http://ontolingua.stanford.edu	--	http://kmi.open.ac.uk/projects/ocml/	http://www.isi.edu/isd/LOOM/LOOM-HOME.html	--
L07	Language_Developer	Stanford University Knowledge Systems Laboratory	--	VITAL project	Southern California ISI	--
L08	Language_Version	6.0	--	--	4.0	--
L09	Last_Version_Date	1997-11-01	--	2001-01-01	2000-05-01	1995-01-01
L1	**DOMAIN KNOWLEDGE (FACTOR)**					
L11	***CONCEPTS/ INSTANCES/FACTS/CLAIMS***					
L111	Allows_Instances_Of_Class	Supported	Supported	Supported	Supported	Supported
L112	Has_Metaclasses	Supported	Supported	Supported	Supported	Supported
L113	Can_Define_Classes_Without_Metaclasses	Supported	Non_supported	Supported	Non_supported	Non_supported
L114	Allows_Facts	Supported	Supported	Supported	Supported	Supported
L115	Allows_Claims	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L12	***ATTRIBUTES***					
L121	Can_Define_Class_Attributes	Supported	Supported	Supported	Supported	Supported
L122	Can_Define_Instance_Attributes	Supported	Supported	Supported	Supported	Supported
L123	Can_Define_Local_Attributes	Supported	Supported	Supported	Supported	Supported
L124	Can_Define_Global_Attributes	Supported	Non_supported	Non_supported	Supported	Non_supported
L125	Can_Define_Polymorph_Attributes	Supported	Supported	Supported	Supported	Non_supported
L126	Can_Define_Exceptions_In_Attributes	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L13	***FACETS***					
L131	Has_Default_Attribute_Values	Supported	Supported	Supported	Supported	Supported
L132	Has_Attribute_Types	Supported	Supported	Supported	Supported	Supported
L133	Can_Define_Cardinality_Of_Attributes	Supported	Supported	Supported	Supported	Supported
L134	Allows_Define_Procedural_Knowledge	Non_supported	Non_supported	Supported	Supported	Non_supported
L135	Allows_New_Facets	Supported	Supported	Non_supported	Supported	Non_supported
L14	***RELATIONS***					
L141	Allows_Definition_Of_Functions	Supported	Non_supported	Supported	Supported	Supported
L142	Arbitrary_N-ary_Relations	Supported	Non_supported	Supported	Supported	Non_supported
L143	Allows_Define_Ad-hoc_Relations	Supported	Supported	Supported	Supported	Supported
L144	Can_Constrain_The_Type_In_Relations	Supported	Supported	Supported	Supported	Supported
L145	Can_Constrain_The_Value_In_Relations	Supported	Non_supported	Supported	Supported	Non_supported
L146	Has_Operational_Definition	Non_supported	Non_supported	Supported	Supported	Supported
L147	Can_Declare_Properties_In_Relations	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L15	***TAXONOMIES***					
L151	Contain_-SubclassOf-_Relation	Supported	Supported	Supported	Supported	Supported
L152	Contain_-NotSubclassOf-_Relation	Supported	Non_supported	Non_supported	Supported	Non_supported
L153	Can_Define_Exhaustive_Decomposition	Supported	Non_supported	Supported	Supported	Supported
L154	Can_Define_Disjoint_ Decomposition	Supported	Non_supported	Supported	Supported	Supported
L155	Multiple-Subclass-of_In_Classes	Supported	Supported	Supported	Supported	Supported
L156	Multiple-Instance-of_In_Instances	Non_supported	Non_supported	Non_supported	Supported	Non_supported
L16	***AXIOMS***					
L161	Allows_Axioms_Embedded_In Terms	Supported	Supported	Supported	Supported	Non_supported
L162	Allows_Independent_Axioms	Supported	Supported	Supported	Non_supported	Non_supported
L163	Allows_Axioms_In_First_Order_Logic	Supported	Supported	Supported	Supported	Supported
L164	Allows_Axioms_In_Second_Order_Logic	Supported	Supported	Non_supported	Non_supported	Non_supported
L17	***PRODUCTION RULES***					
L171	Allows_Disjuntives_In_PRs	Non_supported	Non_supported	Supported	Supported	Non_supported
L172	Allows_Conjuntives_In_PRs	Non_supported	Non_supported	Supported	Supported	Non_supported
L173	Each_Rule_Has_Defined_A_Chaining_Mechanism	Non_supported	Non_supported	Supported	Non_supported	Non_supported
L174	Each_Rule_Has_Defined_A_Priority	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L175	Procedures_In_The_Consequent_In_PRs	Non_supported	Non_supported	Supported	Supported	Non_supported
L176	Certainty_Values_In_PR	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported

Tabla 4.20. Lenguajes clásicos de ontologías: características descriptivas y factor "conocimiento del dominio".

Cód	Característica	DAML+OIL	XOL	SHOE	OIL	OML/CKML
L0	**DESCRIPTIVE CHARACTERISTICS**					
L01	Language_Name	DAML+OIL	XOL	SHOE	OIL	OML/CKML
L02	Type_Of_Language	Markup	Markup	Markup	Markup	Markup
L03	Representation_Formalisms	Frame-based, Description_Logic	Frame-based	Frame-based, Classical_Logic	Frame-based, Description_ Logic	Conceptual _ Graphs
L04	Help-Manuals	High	Low	High	High	High
L05	Allows_Use_Independently_From_Its_Environment	Supported	Supported	Supported	Supported	Supported
L06	Language_Information_URL	http://www.daml.org/	http://www.ai.sri.com/~pkarp/xol/	http://www.cs.umd.edu/ projects/ plus/ SHOE/spec.html	http://www.ontoknowledge .org/oil/	http://www.ontologos.org/OML/OML%200.3.htm
L07	Language_Developer	DAML project	Karp, Chaudhri and Thomere	Maryland University	IST Ontoknowledge project	Kent F.
L08	Language_Version	0.6	0.4	1.01		0.3
L09	Last_Version_Date	2001-12-5	1999-10-01	2000-4-28	2000-11-01	1999-01-01
L1	**DOMAIN KNOWLEDGE (FACTOR)**					
L11	***CONCEPTS/ INSTANCES/ FACTS/ CLAIMS***					
L111	Allows_Instances_Of_Class	Supported	Supported	Supported	Supported	Supported
L112	Has_Metaclasses	Supported	Supported	Non_supported	Non_supported	Non_supported
L113	Can_Define_Classes_Without_Metaclasses	Supported	Supported	Supported	Supported	Non_supported
L114	Allows_Facts	Supported	Supported	Supported	Supported	Supported
L115	Allows_Claims	Non_supported	Non_supported	Supported	Non_supported	Non_supported
L12	***ATTRIBUTES***					
L121	Can_Define_Class_Attributes	Supported	Supported	Non_supported	Supported	Non_supported
L122	Can_Define_Instance_Attributes	Supported	Supported	Supported	Supported	Supported
L123	Can_Define_Local_Attributes	Supported	Supported	Supported	Supported	Supported
L124	Can_Define_Global_Attributes	Supported	Non_supported	Non_supported	Supported	Supported
L125	Can_Define_Polymorph_Attributes	Supported	Non_supported	Non_supported	Supported	Supported
L126	Can_Define_Exceptions_In_Attributes	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L13	***FACETS***					
L131	Has_Default_Attribute_Values	Non_supported	Supported	Non_supported	Non_supported	Supported
L132	Has_Attribute_Types	Supported	Supported	Supported	Supported	Supported
L133	Can_Define_Cardinality_Of_Attributes	Supported	Supported	Non_supported	Supported	Supported
L134	Allows_Define_Procedural_Knowledge	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L135	Allows_New_Facets	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L14	***RELATIONS***					
L141	Allows_Definition_Of_Functions	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L142	Arbitrary_N-ary_Relations	Non_supported	Non_supported	Supported	Non_supported	Supported
L143	Allows_Define_Ad-hoc_Relations	Supported	Non_supported	Supported	Supported	Supported
L144	Can_Constrain_The_Type_In_Relations	Supported	Supported	Supported	Supported	Supported
L145	Can_Constrain_The_Value_In_Relations	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L146	Has_Operational_Definition	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L147	Can_Declare_Properties_In_Relations	Supported	Non_supported	Non_supported	Supported	Non_supported
L15	***TAXONOMIES***					
L151	Contain_-SubclassOf-_Relation	Supported	Supported	Supported	Supported	Supported
L152	Contain_-NotSubclassOf-_Relation	Non_supported	Non_supported	Non_supported	Supported	Supported
L153	Can_Define_Exhaustive_Decomposition	Supported	Non_supported	Non_supported	Non_supported	Non_supported
L154	Can_Define_Disjoint_ Decomposition	Supported	Non_supported	Non_supported	Supported	Non_supported
L155	Multiple-Subclass-of_In_Classes	Supported	Supported	Supported	Supported	Supported
L156	Multiple-Instance-of_In_Instances	Supported	Non_supported	Non_supported	Supported	Non_supported
L16	***AXIOMS***					
L161	Allows_Axioms_Embedded_In Terms	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L162	Allows_Independent_Axioms	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L163	Allows_Axioms_In_First_Order_Logic	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L164	Allows_Axioms_In_Second_Order_Logic	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L17	***PRODUCTION RULES***					
L171	Allows_Disjuntives_In_PRs	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L172	Allows_Conjuntives_In_PRs	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L173	Each_Rule_Has_Defined_A_Chaining_Mechanism	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L174	Each_Rule_Has_Defined_A_Priority	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L175	Procedures_In_The_Consequent_In_PRs	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L176	Certainty_Values_In_PR	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported

Tabla 4.21. Lenguajes de marcado *web* de ontologías: características descriptivas y factor "conocimiento del dominio".

Cód	Característica	Ontolingua	OKBC	OCML	LOOM	FLogic
L2	**INFERENCE MECHANISM (FACTOR)**					
L21	***REASONING POTENTIAL***					
L211	Allows_Multiple_Inheritance	Supported	Supported	Supported	Supported	Supported
L212	Allows_Monotonous_Reasoning	Supported	Non_supported	Supported	Supported	Supported
L213	Allows_Non-Monotonous_Reasoning	Non_supported	Non_supported	Non_supported	Supported	Supported
L214	Makes_Exceptions_In_Inheritance	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L215	Axioms_Keep_The_Consistency	Supported	Supported	Supported	Supported	Supported
L216	Execute_Procedures	Supported	Supported	Supported	Supported	Supported
L217	Inference_Mechanism_In_PR	Non_supported	Non_supported	Supported	Supported	Supported
L22	***INFERENCE ENGINE***					
L221	IE_Is_Sound_and_Complete	Non_supported	Non_supported	Supported	Supported	Supported
L222	IE_Performs_Automatic_Clasifications	Non_supported	Non_supported	Non_supported	Supported	Non_supported
L223	IE_Deals_ Exceptions	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L224	IE_Deals_Multiple_Inherance	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L225	Allows_New_Inference_Engine	Supported	Supported	Supported	Supported	Supported

Tabla 4.22. Lenguajes clásicos de ontologías: características del factor "mecanismo de inferencia".

Cód	Característica	DAML	XOL	SHOE	OIL	OML/CKML
L2	**INFERENCE MECHANISM (FACTOR)**					
L21	***REASONING POTENTIAL***					
L211	Allows_Multiple_Inheritance	Supported	Supported	Supported	Supported	Supported
L212	Allows_Monotonous_Reasoning	Non_supported	Non_supported	Non_supported	Supported	Non_supported
L213	Allows_Non-Monotonous_Reasoning	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L214	Makes_Exceptions_In_Inheritance	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L215	Axioms_Keep_The_Consistency	Supported	Non_supported	Non_supported	Supported	Supported
L216	Execute_Procedures	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L217	Inference_Mechanism_In_PR	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L22	***INFERENCE ENGINE***					
L221	IE_Is_Sound_and_Complete	Non_supported	Non_supported	Supported	Supported	Non_supported
L222	IE_Performs_Automatic_Clasifications	Non_supported	Non_supported	Supported	Supported	Non_supported
L223	IE_Deals_ Exceptions	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L224	IE_Deals_Multiple_Inherance	Non_supported	Non_supported	Non_supported	Supported	Non_supported
L225	Allows_New_Inference_Engine	Supported	Supported	Supported	Supported	Supported

Tabla 4.23. Lenguajes de marcado *web* de ontologías: características del factor "mecanismo de inferencia".

Cód	Característica	Cyc	Uschold & King	Grüninger & Fox	Bernaras et al.	METHONTOLOGY	SENSUS-based	Co4	$(KA)^2$
M1	***PRECISION (FACTOR)***								
M11	Delimitation_Of_Phases	High	Very_Low	Very_High	High	Very_High	High	Medium	Medium
M12	Specification_Of_Activities_By_Phases	Medium	Very_Low	Very_Low	Very_Low	Very_High	Very_Low	Very_Low	Very_Low
M13	Specification_Of_Personnel_By_Phases	Very_Low	Very_Low	Very_Low	Very_Low	Low	Very_Low	Very_Low	Very_Low
M14	Specification_Of_Techiques_By_Phases	Medium	Very_Low	Low	Very_Low	Very_High	Very_Low	Very_Low	Very_Low
M15	Specification_Of_Finished_Products_By_Phases	Very_Low	Very_Low	Very_Low	Very_Low	Very_High	Very_Low	Very_Low	Very_Low
M2	***USABILITY (FACTOR)***								
M21	Clarity_Of_Activities_and_ Techniques _Description	High	High	High	High	High	High	Medium	High
M22	Quality_Of_Manuals	Medium	Very_Low	Very_Low	Very_Low	High	Very_Low	Very_Low	Very_Low
M23	Manuals_With_Complete_Examples	Non_supported	Non_supported	Non_supported	Non_supported	Supported	Non_supported	Non_supported	Non_supported
M3	***MATURITY (FACTOR)***								
M31	Number_Of_Developed_Ontologies	Low	Low	Low	Low	High	Low	Low	Low
M32	Number_Of_Different_Domains	Low	Low	Low	Low	Very_High	Low	Low	High
M33	Importance_Of_Developed_Ontologies	Very_High	High	High	High	High	High	High	Medium

Tabla 4.24. Metodologías de desarrollo de ontologías: características de los factores precisión, usabilidad y madurez.

Cód	Característica	Cyc_methodology	Uschold-King_ methodology	Grüninger-Fox_ methodology	Bernaras_et_al_ methodology	METHONTOLOGY	SENSUS-based_ meth.	Co4_ methodology	$(KA)^2$_ methodology
M01	**Methodology Name**	Cyc_methodology	Uschold-King_ methodology	Grüninger-Fox_ methodology	Bernaras_et_al_ methodology	METHONTOLOGY	SENSUS-based_ meth.	Co4_methodology	$(KA)^2$_methodology
M02	**Methodology Description**	It proposes to codify the explicit and implicit knowledge appearing in the knowledge sources.	This methodology provides guidelines to build ontologies. It proposes to identify the purpose of the ontology, building, evaluating and documentation.	It builds a logic model of knowledges by a formal description of requeriments, whose are obtained from ontology specifications. Then, this description is formalized.	This methodology builds ontologies by integrating and generalizing of knowledge bases.	It is enable the construction of ontologies at the knowledge level. It is inspired in *IEEE Processes*.	It is a methodology used to build ontologies starting from SENSUS abstract concepts.	These KB's are built for being shared, and they have consensual knowledge, so they can be considered ontologies.	It proposes to build ontologies in a joint effort by a group of people at different locations using the same templates and language.
M03	**Based In KE**	Medium	Medium	Low	High	High	Very_Low	Very_Low	Very_Low
M04	**Steps To Build Ontologies**	Codify manually | knowledge codification aided by tools using knowledge already stored at the Cyc KB | codification delegated on the tools the majority of the work	Identification | Ontology Capture | Coding | Integration | Evaluation | Documentation	Capture of scenarios | Formulation of informal competency questions | Specification of terms with formal language | Formulation of formal competency questions | Formal Specification of axioms and terms | Estableshing completeness	Specification of the application | Preliminary design based in top-level categories | Ontology refinement and structuring	Specification | Conceptualization | Formalization | Implementation | Maintenance	Identification of *seed* terms | Linking seed terms to SENSUS |	Proposal | Call for comments | Replay | Decision	Template distribution | Generation of ontologies | Integration | Automated translation
M05	**Recommendations For Knowledge Formalization**	Frame-based, Classical_Logic	None	Classical_Logic	None	None	Semantic-networks	Objects	Objects, Classical_logic
M06	**Strategy For Building Ontologies**	Application-independent	Application-independent	Application-semidependent	Application-dependent	Application-independent	Application-semidependent	Application-independent	Application-independent
M07	**Strategy For Identifying**	Middle-out	Middle-out	Middle-out	Top-down	Middle-out	Down-Top	Not proposed	Not proposed
M08	**Recommended Life Cycle**	Evolving prototypes	None	Not defined totally	None	Evolving prototypes	None	Evolving prototipes	Not defined totally
M09	**Activities At The Life Cycle**	Implementation process | Integral process (Partially)	Requeriment process | Implementation process | Integral process (Partially)	Requeriment process | Design process | Implementation process | Integral process (Partially)	Requeriment process | Design process | Implementation process	Project management processes *Partially* | Requeriment process | Design process | Implementation process | Post-development processes *Partially* | Integral process *Partially*	Requeriment process | Implementation processs	Implementation process | Integral process (partially)	Implementation process
M010	**Recommended Techniques**	Some techniques missing	Unknown	Unknown	Unknown	Some techniques missing	Unknown	Unknown	Unknown
M011	**Technological Support**	Cyc_environment	--	TOVE implem. tool	Unknown	ODE | WebODE	Ontosaurus	Co4 environment	ODE (partially)
M012	**Significant Developed Ontologies**	Cyc KB | Cyc.daml	Enterprise Ontology	TOVE ontology	Ontology for electrical networks	CHEMICALS | Environmental pollutans ontologies | Reference-Ontology | Restrutured $(KA)^2$	SENSUS	Ontologies in molecular genetic domain	Organization-Ontology | Research-Topic-Ontology | …

Tabla 4.25. Metodologías de desarrollo de ontologías: características descriptivas.

Cód	Característica	Ontolingua Editor	WebOnto	OntoEdit
T0	**DESCRIPTIVE CHARACTERISTICS**			
T01	Tool_Name	Ontolingua_Editor	Tadzebao-WebOnto	OntoEdit
T02	Developer_Companies	Stanford University Knowledge Systems Laboratory	The Open University, KMI Institute	Knowledge Management Group of Karlsruhe University, Ontoprise
T03	Description_Of_Sw_Requirements	Internet Explorer or Netscape	Internet Explorer or Netscape	Java 1.3
T04	Description_Of_Hw_Requirements	PC for Internet use	PC for Internet use	PC
T05	Price_Of_Sw-Hw_Requirements	1200	1200	1000
T06	Date_Of_Release	1-1-2001	1-1-2001	3-7-2001
T07	Price_Of_Tool_Licence	0	0	0
T08	Price_Of_Access_Interfaces	0	0	0
T09	Followed_methodology	None	None	None
T010	Translator_Imports_From_Languages	Ontolingua, CML, OKBC, IDL, KIF	OCML	DAML+OIL, RDFS, F-Logic, Prolog
T011	Translator_Exports_To_Languages	Ontolingua, LOOM	OCML, Ontolingua	DAML+OIL, RDFS, F-Logic, Prolog
T012	Ontologies_developed	56 ontologies in several domains	--	--
T1	***CAPABILITIES (FACTOR)***			
T11	Local_Use	Non_Supported	Non_Supported	Supported
T12	Network_Use	Non_Supported	Non_Supported	Non_Supported
T13	Internet-based_Use	Supported	Supported	Non_Supported
T14	Clarity_Of_User_Interface	Low	High	High
T15	Response_Time	Medium	Medium	Very_High
T16	Reliability	Very_High	Medium	Very_Low
T2	***VISUALIZATION (FACTOR)***			
T21	Browsers_Shows_Whole_Information_Of_Terms	Supported	Supported	Supported
T22	Browser_Allows_Selection_Of_Detail_Level	Supported	Supported	Supported
T23	Browser_Shows_Taxonomy	Supported	Supported	Supported
T24	Browser_Shows_Ad-hoc_Relations	Supported	Supported	Supported
T3	***EDITION (FACTOR)***			
T31	Tool_Builds_The_Same_Of_Language	Supported	Supported	Supported
T32	Tool_Allows_Edition_In_Any_Time	Supported	Supported	Supported
T33	Tool_Shows_Taxonomy_Graphically	Supported	Supported	Supported
T34	Tool_Allows_Definition_Of_New_Relations	Non_Supported	Supported	Non_Supported
T4	***INTERACTION (FACTOR)***			
T41	Tool_Allows_Independent_Use	Supported	Supported	Supported
T42	Tool_Supplies_Access_Interfaces	Non_Supported	Non_Supported	Non_Supported
T43	Documentation_Using_Access_Interfaces	Very_Low	Very_Low	Very_Low
T44	Access_Interfaces_Are_OpenSource	Non_Supported	Non_Supported	Non_Supported
T45	Documentation_Programming_Access_Interfaces	Very_Low	Very_Low	Very_Low
T5	***METHODOLOGICAL ASPECTS (FACTOR)***			
T51	Tool_Supports_Whole_Life_Cicle	Non_Supported	Non_Supported	Non_Supported
T52	Tool_Supports_Important_Development_Activities	Supported	Supported	Non_Supported
T53	Tool_Supplies_Documentation_About_Built_Products	Non_Supported	Non_Supported	Non_Supported
T54	Tool_Checks_Consistency	Supported	Supported	Supported
T6	***COOPERATIVE ASPECTS (FACTOR)***			
T61	Tool_Creates_Work_Groups	Supported	Supported	Non_Supported
T62	Tool_Allows_Simultaneous_Working	Supported	Supported	Non_Supported
T63	Tool_Looks_Edited_Ontologies	Supported	Supported	Non_Supported
T64	Tool_Looks_Edited_Terms	Supported	Supported	Non_Supported
T65	Tool_Notifies_The_Changes_To_Group	Supported	Supported	Non_Supported
T66	Tool_Identifies_The_User_Changes	Non_Supported	Non_Supported	Non_Supported
T7	***TRANSLATION (FACTOR)***			
T71	Tool_Imports_From_Others_Langs	Supported	Non_Supported	Supported
T72	Tool_Imports_From_Markup_Langs	Non_Supported	Non_Supported	Supported
T73	Tool_Exports_To_Langs	Non_Supported	Supported	Supported
T74	Tool_Exports_To_Markup_Langs	Non_Supported	Non_Supported	Supported
T75	Translations_Lose_Minimun_Semantic	Non_Supported	Supported	Supported
T76	Translation_Is_Supervised	Non_Supported	Non_Supported	Non_Supported
T8	***INTEGRATION (FACTOR)***			
T81	Ease_Of_Integration	High	High	High
T82	Difficulty_Of_Refering_New_Terms	Low	Low	Low
T83	Tool_Allows_Selection_Of_Terms_To_Integration	Supported	Supported	Non_Supported
T84	Tool_Checks_Consistency_In_Integration_Or_Merge	Supported	Supported	Supported
T85	Assistance_For_Manual_Merge	Non_Supported	Non_Supported	Non_Supported
T86	Semi-automatic_Merge	Non_Supported	Non_Supported	Non_Supported

Tabla 4.26. Entornos de desarrollo de ontologías **(1/2)**.

Cód	Característica	WebODE	OilEd	Protégé2000
T0	**DESCRIPTIVE CHARACTERISTICS**			
T01	Tool_Name	WebODE	OilEd	Protégé2000
T02	Developer_Companies	Knowledge Reuse Group (AI Lab). Universidad Politécnica de Madrid, SPAIN	Univ.Manchester, FreeUniversity of Amsterdam and Interprice GmbH.	Stanford Medical Informatics at the Stanford University School of Medicine
T03	Description_Of_Sw_Requirements	Internet Explorer 5.0 or higher	Java 1.2	Java 1.3
T04	Description_Of_Hw_Requirements	PC for Internet use	PC	PC
T05	Price_Of_Sw-Hw_Requirements	1200	1000	1000
T06	Date_Of_Release	5-5-2001	2-7-2001	1-10-2000
T07	Price_Of_Tool_Licence	0	0	0
T08	Price_Of_Access_Interfaces	0	0	0
T09	Followed_methodology	METHONTOLOGY	None	None
T010	Translator_Imports_From_Languages	CARIN, X-WebODE	DAML, RDFS	F-Logic, OIL, Ontolingua, RDFS
T011	Translator_Exports_To_Languages	CARIN, DAML+OIL, Flogic, OIL, Prolog, X-WebODE	DAML, RDFS, SHIQ, Dotty	F-Logic, OIL, Ontolingua, RDFS
T012	Ontologies_developed	E-commerce ontologies | Knowledge management ontologies | Reference Ontology	$(KA)^2$ Ontology, The CIA World Factbook, 12 Ontologies from the Tourist Domain, OIL in OIL	Biological Processes | Dublin Core | Guideline Interchange Format | Resource-Event-Agent Enterprise | Science Ontology | Semantic Translation | Suggested Upper Merged Ontology
T1	***CAPABILITIES (FACTOR)***			
T11	Local_Use	Non_Supported	Supported	Supported
T12	Network_Use	Non_Supported	Non_Supported	Non_Supported
T13	Internet-based_Use	Supported	Non_Supported	Non_Supported
T14	Clarity_Of_User_Interface	Very_High	High	Very_High
T15	Response_Time	Medium	Very_High	Very_High
T16	Reliability	High	Low	High
T2	***VISUALIZATION (FACTOR)***			
T21	Browsers_Shows_Whole_Information_Of_Terms	Supported	Supported	Supported
T22	Browser_Allows_Selection_Of_Detail_Level	Supported	Supported	Supported
T23	Browser_Shows_Taxonomy	Supported	Supported	Supported
T24	Browser_Shows_Ad-hoc_Relations	Supported	Supported	Supported
T3	***EDITION (FACTOR)***			
T31	Tool_Builds_The_Same_Of_Language	Supported	Non_Supported	Supported
T32	Tool_Allows_Edition_In_Any_Time	Supported	Supported	Supported
T33	Tool_Shows_Taxonomy_Graphically	Supported	Supported	Supported
T34	Tool_Allows_Definition_Of_New_Relations	Supported	Non_Supported	Supported
T4	***INTERACTION (FACTOR)***			
T41	Tool_Allows_Independent_Use	Supported	Supported	Supported
T42	Tool_Supplies_Access_Interfaces	Non_Supported	Non_Supported	Supported
T43	Documentation_Using_Access_Interfaces	Very_Low	Very_Low	High
T44	Access_Interfaces_Are_OpenSource	Non_Supported	Non_Supported	Supported
T45	Documentation_Programming_Access_Interfaces	Very_Low	Very_Low	High
T5	***METHODOLOGICAL ASPECTS (FACTOR)***			
T51	Tool_Supports_Whole_Life_Cicle	Non_Supported	Non_Supported	Non_Supported
T52	Tool_Supports_Important_Development_Activities	Supported	Non_Supported	Supported
T53	Tool_Supplies_Documentation_About_Built_Products	Non_Supported	Non_Supported	Supported
T54	Tool_Checks_Consistency	Supported	Supported	Supported
T6	***COOPERATIVE ASPECTS (FACTOR)***			
T61	Tool_Creates_Work_Groups	Supported	Non_Supported	Non_Supported
T62	Tool_Allows_Simultaneous_Working	Supported	Non_Supported	Non_Supported
T63	Tool_Looks_Edited_Ontologies	Supported	Non_Supported	Non_Supported
T64	Tool_Looks_Edited_Terms	Non_Supported	Non_Supported	Non_Supported
T65	Tool_Notifies_The_Changes_To_Group	Non_Supported	Non_Supported	Non_Supported
T66	Tool_Identifies_The_User_Changes	Non_Supported	Non_Supported	Non_Supported
T7	***TRANSLATION (FACTOR)***			
T71	Tool_Imports_From_Others_Langs	Supported	Non_Supported	Supported
T72	Tool_Imports_From_Markup_Langs	Supported	Supported	Supported
T73	Tool_Exports_To_Langs	Supported	Non_Supported	Supported
T74	Tool_Exports_To_Markup_Langs	Supported	Supported	Supported
T75	Translations_Lose_Minimun_Semantic	Supported	Non_Supported	Non_Supported
T76	Translation_Is_Supervised	Non_Supported	Non_Supported	Non_Supported
T8	***INTEGRATION (FACTOR)***			
T81	Ease_Of_Integration	Very_Low	Very_Low	High
T82	Difficulty_Of_Refering_New_Terms	Very_Low	Very_Low	Low
T83	Tool_Allows_Selection_Of_Terms_To_Integration	Non_Supported	Non_Supported	Supported
T84	Tool_Checks_Consistency_In_Integration_Or_Merge	Non_Supported	Non_Supported	Supported
T85	Assistance_For_Manual_Merge	Non_Supported	Non_Supported	Non_Supported
T86	Semi-automatic_Merge	Non_Supported	Non_Supported	Non_Supported

Tabla 4.26. Entornos de desarrollo de ontologías **(2/2)**.

4.6. SELECCIÓN DE ONTOLOGÍAS CANDIDATAS

Identificado el conjunto de dimensiones, factores y características que describen el dominio de las ontologías, y las instancias de cada una de las dimensiones, la siguiente tarea consiste en seleccionar las ontologías que van a ser consideradas como candidatas para ser usadas en el proyecto. Este conjunto de ontologías candidatas estará formado por todas aquellas ontologías que satisfagan los requisitos establecidos por el usuario, según las necesidades de su sistema. Por ejemplo sería ilógico calcular la idoneidad de una ontología cuyos términos no coincidan, o cuyo paradigma de representación del conocimiento sea distinto al seleccionado en el proyecto, o que estuviera implementada en un lenguaje con capacidades de representación y razonamiento diferentes a los requeridos, o cuyos costes de uso superen el presupuesto del proyecto, o con una metodología que no tenga un nivel de usabilidad mínimo exigido, etc.

La manera de exigir estos requisitos es indicando, en el AMC, los valores obligatorios que deben tener las ontologías candidatas en las correspondientes características, situadas en una o varias dimensiones. Para cada una de las ontologías almacenadas en la RO se deberá comprobar si cumplen los valores indicados para ser consideradas como candidatas.

4.6.1. Herramienta Software para la Selección de Ontologías: *Reference Ontology Instances Selector*

Llevar a cabo de forma "manual" la selección de ontologías sería una labor muy tediosa al tener que analizar, una a una, las características de las cinco dimensiones de las ontologías almacenadas en la RO. Por ello, se ha desarrollado una aplicación software que asiste en el proceso de selección de ontologías candidatas, que se ha llamado ***Reference Ontology Instances Selector*** (ROIS). ROIS no sólo sirve para seleccionar las ontologías candidatas que van a ser analizadas por el método *OntoMetric*, sino que puede ser utilizada, de forma independiente, como una completa herramienta de búsqueda en el dominio de ontologías, al permitir buscar también de forma independiente lenguajes, metodologías, y entornos de desarrollo de ontologías que satisfagan ciertos requisitos. Está accesible en Internet y puede ser ejecutada desde cualquier navegador *web*.

La aplicación software ROIS, como se representa en la figura 4.7, consta de una serie de módulos para realizar el proceso de selección de ontologías candidatas:

- **Constructor del AMC**: toma el modelo conceptual de la ontología RO, exportado desde la aplicación *WebODE* (en formato *X-WebODE*), y lo transforma en un conjunto dinámico de formularios de consulta, que representan el AMC.

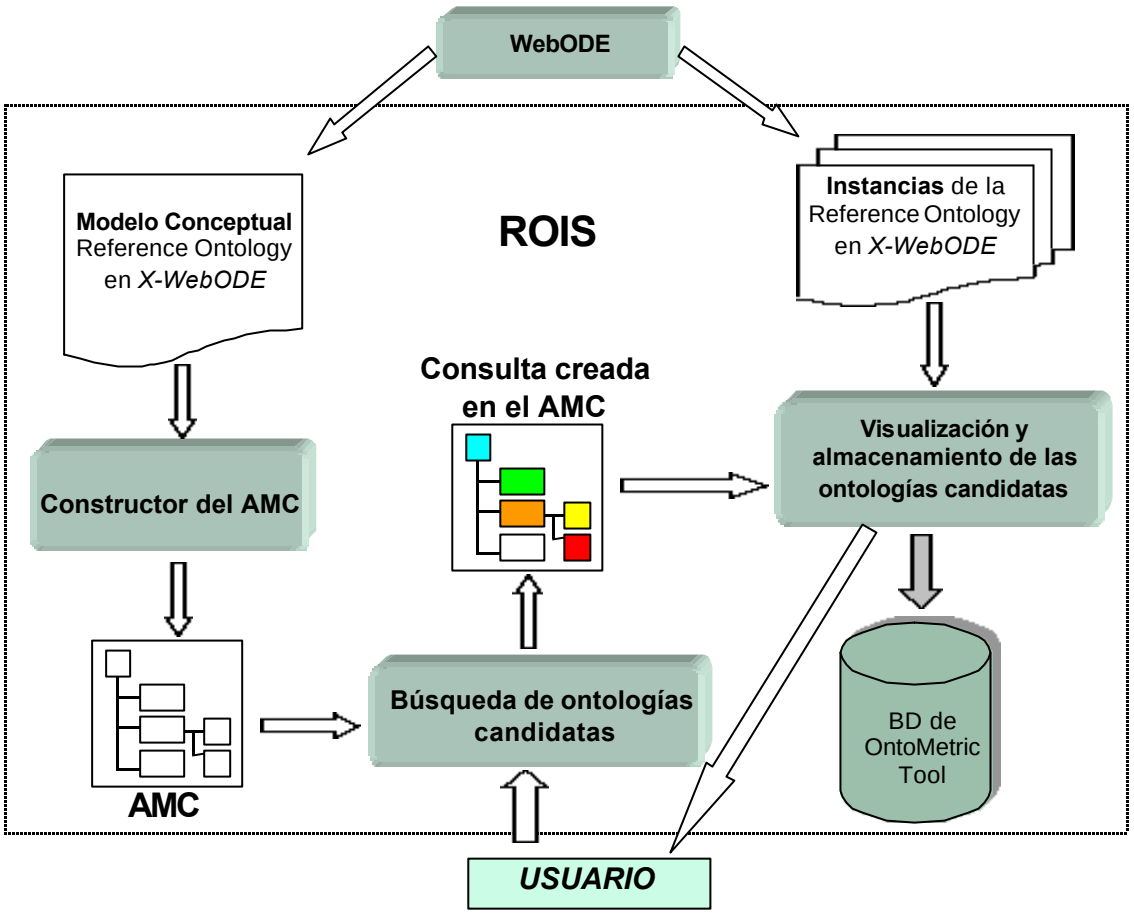


Figura 4.7. Arquitectura de ROIS.

- **Búsqueda de ontologías candidatas**: El usuario podrá navegar por el conjunto de formularios, profundizando en el grado de detalle deseado, e indicando en las características los valores que deben tomar cada una de ellas. La selección puede restringirse, a su vez, en una o varias de las dimensiones del AMC. ROIS realiza el proceso de búsqueda en el conjunto de instancias de la RO.

- **Visualización y almacenamiento de información relacionada con las ontologías candidatas**. El proceso presenta por pantalla al usuario las instancias de las ontologías que han satisfecho la consulta en las dimensiones y niveles seleccionados. Muestra toda la información disponible relacionada con el "contenido" de las ontologías candidatas y los "costes de uso". Y almacena también la información sobre los "lenguajes", "metodologías" y "entornos", relacionados con las ontologías candidatas en una base de datos relacional que es importable desde la aplicación *OntoMetric Tool.*

4.7. MEDICIÓN DE IDONEIDAD DE ONTOLOGÍAS CANDIDATAS

Las secciones 4.4, 4.5 y 4.6 han descrito: el marco multinivel de características y la construcción de la RO, cómo se ha instanciado la RO, y la aplicación ROIS que asiste al proceso de selección de ontologías candidatas. No obstante, queda aún por resolver: ¿cómo saber cuáles de estas ontologías candidatas son las más idóneas?, ¿cómo se pueden comparar? y ¿cómo se puede justificar la elección de unas frente a otras?

En esta sección se describe el proceso que propone *OntoMetric* para resolver estas cuestiones. En el apartado 4.7.1 se describe el método AHP en el caso de la elección de ontologías. Y en el apartado 4.7.2. se muestran las tareas que realiza la aplicación OT, que asiste en la aplicación de este proceso.

4.7.1. El Método de las Jerarquías Analíticas Adaptado a la Elección de Ontologías

Como se indicó en el apartado 2.6.4, el método AHP se utiliza en procesos de decisión multicriterio en los que se evalúa la idoneidad de varias alternativas frente a un objetivo. El proceso de elegir ontologías para usarlas en un proyecto pertenece a este tipo de problemas, pues el **objetivo** es escoger las ontologías más idóneas según las necesidades de un determinado proyecto, las **alternativas** son las diferentes ontologías disponibles de un dominio determinado, y la **idoneidad** es obtenida a partir de múltiples criterios que serán ponderados según las necesidades del proyecto.

El método *OntoMetric* se basa en los pasos y procesos propuestos por AHP. A continuación se presentan los pasos del método AHP adaptado a la elección de ontologías:

PASO 1: Establecer el problema y sobre qué objetivos se va a tomar la decisión. Para el objetivo que se persigue en el método *OntoMetric*, el problema es seleccionar las ontologías más idóneas para incorporarlas a un proyecto, y los objetivos estarán marcados por las directrices de este proyecto.

PASO 2: Organizar el árbol jerárquico de criterios. En *OntoMetric*, los criterios son las dimensiones, factores y características que componen el marco multinivel de características, descritos en la sección 4.4. El árbol jerárquico de criterios, para el caso de uso de ontologías,

es el AMC descrito en el apartado 4.4.1.2. El AMC deberá ser ajustado, como se describe en el apartado 4.7.1.1, conforme a los objetivos identificados en el paso 1.

PASO 3: Cálculo de pesos de importancia entre criterios hermanos del árbol jerárquico de criterios. El usuario que utilice el método *OntoMetric* puede asignar los pesos de importancia a cada nodo del AMC de dos formas:

1) A partir de las matrices de comparación de criterios, tal y como propone AHP, y que se describe en el apartado 4.7.1.2.1, o bien,

2) Tomando los pesos por defecto, que se han obtenido mediante encuestas de opinión realizadas a expertos en ontologías, mostrados en el apartado 4.7.1.2.2.

PASO 4: Valoración de los nodos hoja del árbol de decisión. El método *OntoMetric* propone unas escalas de valoración lingüística para valorar los nodos hoja del AMC, que se describe en el apartado 4.7.1.3.1. La valoración de las dimensiones del AMC, como se describe en el apartado 4.7.1.3.2, también se realiza de dos formas:

1) Para las dimensiones de "lenguajes", "metodologías" y "entornos" se realiza de forma directa a partir de los valores de las instancias almacenadas en la RO.

2) Para las dimensiones de "contenido" y "costes" el usuario debe asignar los valores de los nodos hoja del AMC a partir de la información existente en la RO en estas dimensiones.

PASO 5: Cálculo de los valores de idoneidad. Para cada ontología candidata se realiza una suma ponderada que establecerá la medida de idoneidad de cada ontología a las necesidades del proyecto. En el apartado 4.7.1.4 se describe cómo se obtienen estas medidas, y cómo se comparan los resultados obtenidos.

Finalmente, en el apartado 4.7.1.5 se resumen cuáles son las diferencias de *OntoMetric* con el método AHP. A continuación se detallan los pasos del 2 al 5.

4.7.1.1. Paso 2: Organización del Árbol Jerárquico de Criterios

Como ya se ha comentado, *OntoMetric* toma como árbol jerárquico de criterios el AMC presentado en la sección 4.4.1.2 de este trabajo, y los criterios son las dimensiones, factores y características identificados en el marco multinivel de características. Considerando como documento de trabajo los objetivos y directrices establecidos para el proyecto, el usuario debe modificar el AMC para definir cada uno de los criterios de decisión con el grado de detalle requerido.

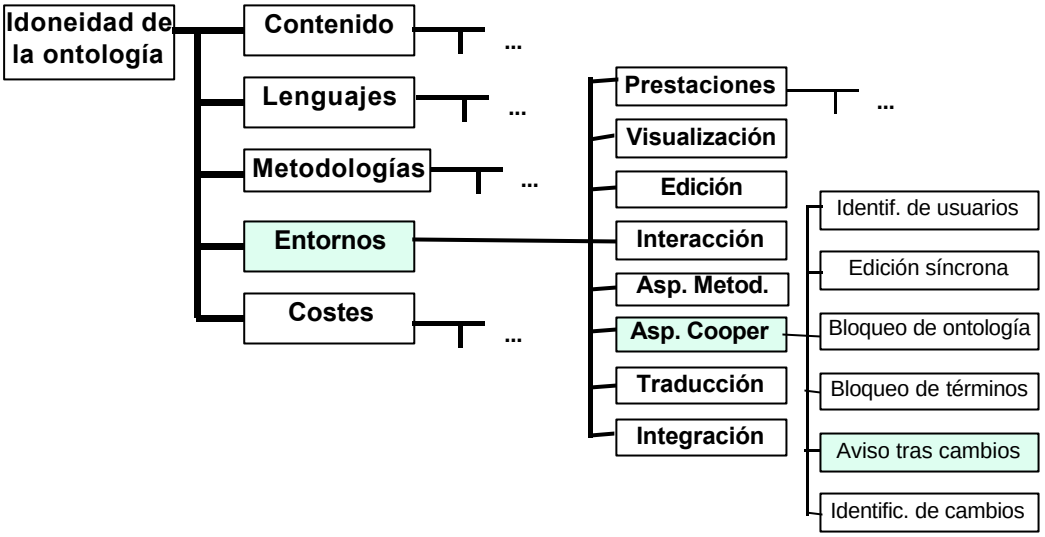


Figura 4.8. Expansión del AMC para incluir características que permitan representar: "El entorno debe notificar automáticamente al equipo de desarrollo los cambios producidos en los términos de la ontología".

El proceso de ajuste del AMC debe realizarse de forma que las características del último nivel del AMC escogido, para un criterio determinado, recojan todas las directrices establecidas para el proyecto. Evidentemente, la exactitud de la medición final, y también la justificación que pueda hacerse de la decisión tomada, va a depender de los niveles que se profundice en el árbol.

Por ejemplo, si una de las directrices que se establece en el proyecto indica que: "Es obligatorio que cuando se realice una modificación en algún término de la ontología se notifique, de forma automática, a todos los miembros del equipo de desarrollo del proyecto", el usuario debe buscar en la dimensión "entornos de desarrollo", y más concretamente en el factor "aspectos cooperativos", si la característica "Los entornos software avisan de los cambios producidos a los componentes del grupo" está definida.

Si se necesita profundizar en el AMC para que aparezca algún criterio determinado (como en el ejemplo anterior) que recoja alguna directriz establecida en el proyecto, en este nuevo nivel deben aparecer también los criterios hermanos del criterio incluido, ya que éstos describen en conjunto al criterio padre ampliado. Siguiendo con el ejemplo anterior, como se muestra en la figura 4.8, si se necesita que aparezca la característica "las herramientas avisan de los cambios producidos a los componentes del grupo", expandiendo el factor "aspectos cooperativos", en el árbol deben aparecer las características "hermanas" de la característica anterior (*"identificación de los usuarios"*, *"edición síncrona"*, etc.), por tener influencia conjunta en el valor del factor "aspectos cooperativos".

También puede ocurrir lo contrario; es decir, criterios existentes en el AMC que no aparecen en la definición de objetivos del proyecto. En este caso, el usuario debe decidir si los mantiene o poda las ramas de esos criterios. De cualquier forma, una vez establecidos y fijados los criterios

del AMC, estos serán siempre los mismos en el proceso de valoración de la idoneidad de todas las ontologías candidatas.

4.7.1.2. Paso 3: Ponderación de los Criterios del Árbol Multinivel de Características

En este proceso, el usuario deberá indicar, atendiendo a las directrices y objetivos del proyecto, la importancia que se le asigna a cada uno de los criterios del AMC. O puede utilizar, como base para realizar su medición, el conjunto de pesos por defecto asignados a los criterios del AMC que proporciona *OntoMetric*.

En cualquiera de los dos casos, el resultado del proceso de ponderación será el AMC con un peso normalizado asignado a cada uno de los nodos del árbol; esto es, que los pesos de los nodos hermanos (hijos de un mismo nodo padre) sumarán 1. Este peso reflejará la importancia que tiene, para un determinado proyecto, cada uno de los nodos frente a sus nodos hermanos. Para abreviar, se usarán las siglas AMCP para hacer referencia a este **árbol multinivel de características ponderado**.

El método AHP propone asignar los pesos de importancia haciendo comparaciones, dos a dos, entre criterios hermanos mediante una matriz de comparación de criterios o también llamada matriz de comparación de pares. Una vez que el usuario haya hecho todas las asignaciones de importancia de los criterios, deberá calcularse el ***autovector*** de pesos de la matriz, que denotará la contribución de cada criterio al nodo padre. Estos conceptos y procesos serán descritos a continuación.

4.7.1.2.1. Obtención de los Pesos de Importancia a partir de las Matrices de Comparación de Criterios

Una vez adaptado el AMC a las necesidades del proyecto, el usuario debe, teniendo en cuenta también estas especificaciones, determinar la importancia que tiene cada uno de los criterios (en relación con sus nodos hermanos) en el AMC. En este paso, se explica cómo en AHP se determina la importancia de los criterios mediante las **matrices de comparación de pares**.

Para llevar a cabo este paso, debe construirse una matriz de comparación por cada uno de los nodos del AMC que tenga algún conjunto de nodos hijos. Usando como ejemplo, para seguir las fases de este proceso, la construcción de una matriz de comparación para el factor "usabilidad" de la dimensión "metodología" a partir del AMC expuesto en la figura 4.9.

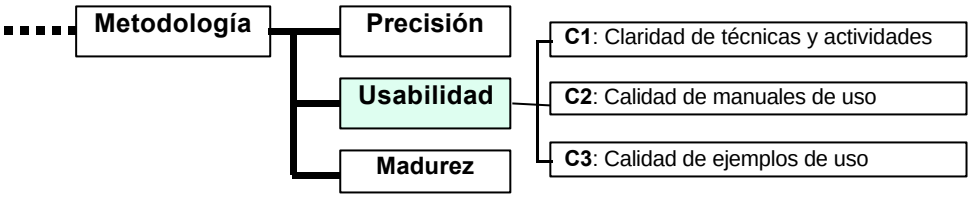


Figura 4.9. Parte del AMC con el factor "usabilidad".

Las matrices de comparación se construyen de la siguiente forma:

a) Se construye una matriz, como la de la figura 4.10, cuyas filas y columnas sean el conjunto de nodos hijos del criterio que se quiere ponderar, y cuya diagonal tenga el valor 1. En el ejemplo sería:

Usabilidad	C1	C2	C3
C1	1		
C2		1	
C3			1

Figura 4.10. Ejemplo de matriz de comparación.

b) Para cada par de criterios, se determina la importancia relativa que tiene un criterio frente al otro, a partir de la siguiente **escala de referencia**: (1) igual de importante que, (3) moderadamente más importante que, (5) más importante que, (7) mucho más importante que, y (9) extremadamente más importante que. Los valores (2), (4), (6) y (8) son los juicios de importancia intermedia de esta escala.

Para cada elemento (i,j) de la matriz, siendo i diferente de j, se debe indicar qué importancia tiene para el proyecto la característica i frente a la característica j. Se comprueba que la matriz está bien construida confirmando que el elemento (i,j) es el inverso matemático del elemento (j,i), es decir: $(i,j) = (1 / (j,i))$

En el ejemplo se supone que, para el proyecto, el usuario ha considerado, analizando las directrices del proyecto, que:

- "La característica C2 es moderadamente más importante (3) que la característica C1".
- "La característica C2 es entre igual y moderadamente más importante que (2) la característica C3".
- "La característica C1 es más importante que (5) la característica C3".

Esto se representa en la matriz de comparación como la que aparece en la figura 4.11:

Usabilidad	C1	C2	C3
C1	1	1/3	5
C2	3	1	2
C3	1/5	1/2	1

Figura 4.11. Ejemplo de valoración de importancia en la matriz de comparación.

c) Se hallan los autovectores (también denominados *eigenvectores*) de todas las matrices de comparación de pares. El autovector resultante representará el grado de importancia de cada criterio frente a sus hermanos. Una forma de resolver el autovector [Saa90] es elevándolo al cuadrado repetidamente, hasta que la suma de sus columnas normalizadas no varíe del paso anterior más de un valor definido (en [Saa90] se aconsejan cuatro decimales). La misma fórmula se ha seguido para implementar este proceso en la aplicación OT. En el ejemplo, el autovector resultante sería:

C1: 0.3420 C2: 0.5241 C3: 0.1339

A partir de estos valores se puede decir que la característica C2 sería la característica más importante, C1 la siguiente, y C3 la menos importante, y además en la proporción que indican los pesos calculados mediante el autovector.

Los pesos hallados a partir de los autovectores deben asociarse en el AMCP a cada criterio correspondiente, como en el ejemplo de la figura 4.12. El resultado final de este proceso es el AMCP, en el que todos los nodos tienen un peso de importancia asociado.

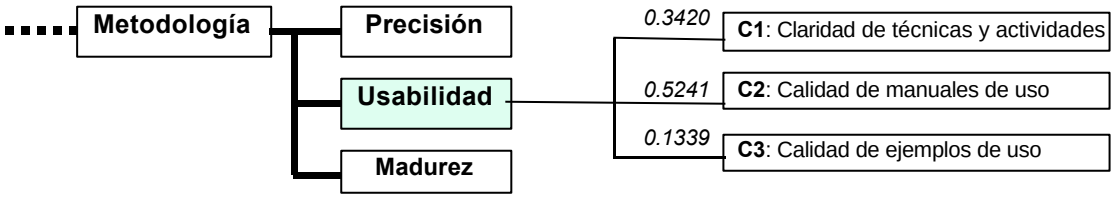


Figura 4.12. Parte del AMC con los pesos asociados a las características del factor "usabilidad".

4.7.1.2.2. Pesos por Defecto Asignados al Árbol Multinivel de Características

El método *OntoMetric* ofrece al usuario no experimentado, o que quiera disponer de una valoración inicial de importancia para los criterios, un AMCP por defecto. Los pesos asignados a los nodos del AMCP han sido obtenidos a partir de las apreciaciones de importancia que 10 expertos, familiarizados en el desarrollo de aplicaciones basadas en ontologías, han asignado mediante una encuesta de opinión. Aunque cada proyecto concreto tendrá unas prioridades específicas para decidir elegir unas ontologías frente a otras, la encuesta refleja las valoraciones de importancia usuales que suelen tomar los criterios de decisión en cualquier tipo de proyecto.

A partir de las encuestas de opinión se han obtenido los pesos de los nodos del AMCP realizando la media aritmética sobre las apreciaciones de los encuestados. Los pesos por defecto obtenidos son los que aparecen en las tablas 4.27, 4.28, 4.29, 4.30 y 4.31. Estos pesos son asignados únicamente a las "características usadas para la medición de idoneidad" de cada una de las dimensiones, ya que las características descriptivas, como se comentó en el apartado 4.4.1.1, sólo son utilizadas para la selección de candidatos. El formulario de la encuesta y los valores marcados por los encuestados, usados para calcular los pesos del AMCP por defecto, se exponen en los anexos I y III de esta memoria.

Cód	Nombre de la característica DIMENSIÓN: CONTENIDO	Peso por defecto 0.2241
	CONCEPTS (FACTOR)	**0,2164**
C11	Essential_Concepts	0,1694
C12	Essential_Concepts_In_Superior_Levels	0,124
C13	Concepts_Properly_described_In_NL	0,1868
C14	Formal_Specification_Of_Concepts_Coincides_With_NL	0,2135
C15	Attributes_Describe_Concepts	0,1893
C16	Number_Of_Concepts	0,1162
	RELATIONS (FACTOR)	**0,2637**
C21	Essential_Relations	0,1464
C22	Relations_Relate_Appropriate_Concepts	0,1249
C23	Formal_Specification_Of_Relations_Coincides_With_NL	0,1831
C24	Arity_Specified	0,1600
C25	Formal_Properties_Of_Relations	0,1077
C26	Number_Of_Relations	0,1005
	TAXONOMY (FACTOR)	**0,2870**
C31	Several_Perspectives	0,2302
C32	Appropriate_Not-Subclass-Of	0,1180
C33	Appropriate_Exhaustive-partitions	0,1983
C34	Appropriate_Disjoint-partitions	0,2054
C35	Maximum_Depth	0,1151
C36	Average_Of_Subclasses	0,1328
	AXIOMS (FACTOR)	**0,2327**
C41	Axioms_Solve_Queries	0,1959
C42	Axioms_Infer_Knowledge	0,2239
C43	Axioms_Verify_Consistency	0,2849
C44	Axioms_Not_Linked_To_Concepts)	0,1526
C45	Number_Of_Axioms	0,1424

Tabla 4.27. Pesos por defecto asignados a los criterios de la dimensión "contenido".

Cód	Nombre de la característica	Peso por defecto
	DIMENSIÓN: LENGUAJES	**0.2275**
	DOMAIN KNOWLEDGE (FACTOR)	**0,5454**
	CONCEPTS/ INSTANCES/ FACTS/ CLAIMS	**0,1658**
L111	Allows_Instances_Of_Class	0,3172
L112	Has_Metaclasses	0,244
L113	Can_Define_Classes_Without_Metaclasses	0,163
L114	Allows_Facts	0,1903
L115	Allows_Claims	0,0845
	ATTRIBUTES	**0,1512**
L121	Can_Define_Class_Attributes	0,1905
L122	Can_Define_Instance_Attributes	0,2223
L123	Can_Define_Local_Attributes	0,1488
L124	Can_Define_Global_Attributes	0,1225
L125	Can_Define_Polymorph_Attributes	0,1548
L126	Can_Define_Exceptions_In_Attributes	0,1608
	FACETS	**0,1512**
L131	Has_Default_Attribute_Values	0,1867
L132	Has_Attribute_Types	0,2334
L133	Can_Define_Cardinality_Of_Attributes	0,2267
L134	Allows_Define_Procedural_Knowledge	0,1778
L135	Allows_New_Facets	0,1752
	RELATIONS	**0,1396**
L141	Allows_Definition_Of_Functions	0,1417
L142	Arbitrary_N-ary_Relations	0,1316
L143	Allows_Define_Ad-hoc_Relations	0,1975
L144	Can_Constrain_The_Type_In_Relations	0,1253
L145	Can_Constrain_The_Value_In_Relations	0,1692
L146	Has_Operational_Definition	0,1432
L147	Can_Declare_Properties_In_Relations	0,0911
	TAXONOMIES	**0,1512**
L151	Contain_-SubclassOf-_Relation	0,2079
L152	Contain_-NotSubclassOf-_Relation	0,1224
L153	Can_Define_Exhaustive_Decomposition	0,1559
L154	Can_Define_Disjoint_ Decomposition	0,1507
L155	Multiple-Subclass-of_In_Classes	0,1812
L156	Multiple-Instance-of_In_Instances	0,1812
	AXIOMS	**0,1309**
L161	Allows_Axioms_Embedded_In Terms	0,2925
L162	Allows_Independent_Axioms	0,1608
L163	Allows_Axioms_In_First_Order_Logic	0,2778
L164	Allows_Axioms_In_Second_Order_Logic	0,2687
	PRODUCTION RULES	**0,109**
L171	Allows_Disjuntives_In_PRs	0,1716
L172	Allows_Conjuntives_In_PRs	0,1144
L173	Each_Rule_Has_Defined_A_Chaining_Mechanism	0,1601
L174	Each_Rule_Has_Defined_A_Priority	0,1830
L175	Procedures_In_The_Consequent_In_PRs	0,1784
L176	Certainty_Values_In_PR	0,1922
	INFERENCE MECHANISM (FACTOR)	**0,4545**
	REASONING POTENTIAL	**0,5157**
L211	Allows_Multiple_Inheritance	0,1811
L212	Allows_Monotonous_Reasoning	0,1314
L213	Allows_Non-Monotonous_Reasoning	0,1183
L214	Makes_Exceptions_In_Inheritance	0,1652
L215	Axioms_Keep_The_Consistency	0,1709
L216	Execute_Procedures	0,1427
L217	Inference_Mechanism_In_PR	0,0901
	INFERENCE ENGINE	**0,4842**
L221	IE_Is_Sound_and_Complete	0,1943
L222	IE_Performs_Automatic_Clasifications	0,1943
L223	IE_Deals_ Exceptions	0,1873
L224	IE_Deals_Multiple_Inherance	0,2366
L225	Allows_New_Inference_Engine	0,1873

Tabla 4.28. Pesos por defecto asignados a los criterios de la dimensión “lenguajes”.

Cód	Nombre de la característica DIMENSIÓN: METODOLOGÍA	Peso por defecto 0.1615
	PRECISION (FACTOR)	**0,3089**
M11	Delimitation_Of_Phases	0,1871
M12	Specification_Of_Activities_By_Phases	0,2014
M13	Specification_Of_Personnel_By_Phases	0,1942
M14	Specification_Of_Techiques_By_Phases	0,2086
M15	Specification_Of_Finished_Products_By_Phases	0,2086
	USABILITY (FACTOR)	**0,3707**
M21	Clarity_Of_Activities_and_ Techniques _Description	0,3311
M22	Quality_Of_Manuals	0,3398
M23	Manuals_With_Complete_Examples	0,3291
	MATURITY (FACTOR)	**0,3204**
M31	Number_Of_Developed_Ontologies	0,2967
M32	Number_Of_Different_Domains	0,3407
M33	Importance_Of_Developed_Ontologies	0,3626

Tabla 4.29. Pesos por defecto asignados a los criterios de la dimensión “metodología”.

Cód	Nombre de la característica DIMENSIÓN: ENTORNOS	Peso por defecto 0.2483
	CAPABILITIES (FACTOR)	**0,1316**
T11	Local_Use	0,1316
T12	Network_Use	0,1283
T13	Internet-based_Use	0,1332
T14	Clarity_Of_User_Interface	0,2122
T15	Response_Time	0,1875
T16	Reliability	0,2072
	VISUALIZATION (FACTOR)	**0,1316**
T21	Browsers_Shows_Whole_Information_Of_Terms	0,2651
T22	Browser_Allows_Selection_Of_Detail_Level	0,241
T23	Browser_Shows_Taxonomy	0,253
T24	Browser_Shows_Ad-hoc_Relations	0,241
	EDITION (FACTOR)	**0,1316**
T31	Tool_Builds_The_Same_Of_Language	0,2723
T32	Tool_Allows_Edition_In_Any_Time	0,264
T33	Tool_Shows_Taxonomy_Graphically	0,2374
T34	Tool_Allows_Definition_Of_New_Relations	0,2263
	INTERACTION (FACTOR)	**0,1177**
T41	Tool_Allows_Independent_Use	0,1901
T42	Tool_Supplies_Access_Interfaces	0,2068
T43	Documentation_Using_Access_Interfaces	0,218
T44	Access_Interfaces_Are_OpenSource	0,1801
T45	Documentation_Programming_Access_Interfaces	0,205
	METHODOLOGICAL ASPECTS (FACTOR)	**0,1070**
T51	Tool_Supports_Whole_Life_Cicle	0,1832
T52	Tool_Supports_Important_Development_Activities	0,2623
T53	Tool_Supplies_Documentation_About_Built_Products	0,2548
T54	Tool_Checks_Consistency	0,2998
	COOPERATIVE ASPECTS (FACTOR)	**0,1156**
T61	Tool_Creates_Work_Groups	0,1711
T62	Tool_Allows_Simultaneous_Working	0,1467
T63	Tool_Looks_Edited_Ontologies	0,1418
T64	Tool_Looks_Edited_Terms	0,1847
T65	Tool_Notifies_The_Changes_To_Group	0,1711
T66	Tool_Identifies_The_User_Changes	0,1847
	TRANSLATION (FACTOR)	**0,1364**
T71	Tool_Imports_From_Others_Langs	0,1733
T72	Tool_Imports_From_Markup_Langs	0,1841
T73	Tool_Exports_To_Langs	0,1492
T74	Tool_Exports_To_Markup_Langs	0,1733
T75	Translations_Lose_Minimun_Semantic	0,1841
T76	Translation_Is_Supervised	0,1361
	INTEGRATION (FACTOR)	**0,1284**
T81	Ease_Of_Integration	0,1595
T82	Difficulty_Of_Refering_New_Terms	0,1652
T83	Tool_Allows_Selection_Of_Terms_To_Integration	0,1823
T84	Tool_Checks_Consistency_In_Integration_Or_Merge	0,1924
T85	Assistance_For_Manual_Merge	0,1538
T86	Semi-automatic_Merge	0,1468

Tabla 4.30. Pesos por defecto asignados a los criterios de la dimensión “entornos”.

Cód	Nombre de la característica DIMENSIÓN: COSTES	Peso por defecto 0.1438
Cost1	Use_Licences_of_the_Ontology (FACTOR)	0,2537
Cost2	Estimated_costs_of_hw_and_sw (FACTOR)	0.2537
Cost3	Costs_of_access_interfaces (FACTOR)	0.2562
Cost4	Use_Licences_of_the_ontology_tools (FACTOR)	0.2562

Tabla 4.31. Pesos por defecto asignados a los criterios de la dimensión "costes".

A pesar de que estos pesos por defecto proporcionan una valoración inicial de la importancia de los criterios, una opción más recomendable sería utilizar como base un AMCP de algún proyecto que pertenezca al mismo dominio [Bas94] del proyecto que se va a desarrollar, ya que las apreciaciones sobre la importancia de los criterios estarían más próximas a los AMCP de proyectos similares. El usuario debería tener en cuenta si su empresa ha realizado un proceso de valoración para un proyecto del mismo dominio, o si puede obtener un AMCP desde otras fuentes.

4.7.1.3. Paso 4: Valoración de los Nodos Hoja del Árbol Multinivel de Características para las Ontologías Candidatas

En este proceso, para cada una de las ontologías candidatas, deberá crearse un AMCP cuyos nodos hoja contengan la valoración que en esos criterios tenga la ontología analizada. Para realizar esta valoración, *OntoMetric* usa valores lingüísticos que son asignados a los nodos hoja del AMCP, de forma que el usuario puede apreciar de forma más intuitiva estas valoraciones.

La asignación de los valores a los nodos hoja del AMC se hace de dos formas diferentes, dependiendo de la dimensión que se esté valorando:

a) Para las dimensiones de "lenguajes", "metodologías" y "entornos", la asignación se realiza directamente utilizando los valores de las instancias almacenadas en la RO.

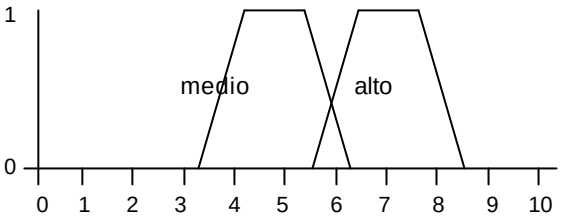
b) Para las dimensiones de "contenido" y "costes", la valoración la realiza el usuario a partir de la información mostrada por ROIS con información de las instancias almacenadas en la RO.

4.7.1.3.1. Valores Asociados a Escalas Lingüísticas

Para hacer el cálculo de idoneidad de cada ontología, cada criterio que aparece como nodo hoja en el AMCP debe tener un valor asignado. Este valor indicará el grado de adecuación que tiene la ontología en ese criterio determinado, respecto al proyecto estudiado.

En el método *OntoMetric*, estos criterios tomarán valores discretos y pertenecientes a una

muy bajo	0	0	1.2	2.2
bajo	1.2	2.2	3.4	4.4
medio	3.4	4.4	5.6	6.6
alto	5.6	6.6	7.8	8.8
muy alto	7.8	8.8	10	10



No soportado	0	0	3.3	6.6
Soportado	3.3	6.6	10	10

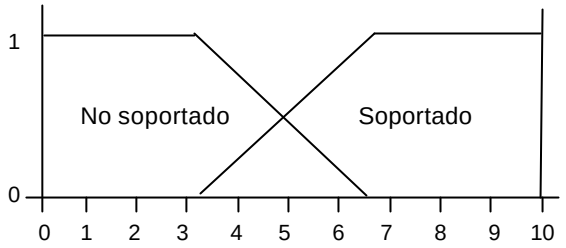


Figura 4.13. Ejemplos de identificación de valores lingüísticos con intervalos difusos.

escala fijada para cada una de los nodos del AMCP; pero el usuario podrá elegir otra escala de valoración si desea asignar mayor o menor detalle a los valores que puede tomar ese criterio.

De forma diferente a como valora las alternativas el método AHP, *OntoMetric* utiliza **escalas lingüísticas de valoración** [Gom97] para cada criterio del AMCP. Con ello se busca que el procedimiento de asignar el valor a un criterio sea más intuitivo, ya que los humanos suelen sentirse más cómodos al realizar calificaciones lingüísticas en vez de numéricas, en apreciaciones de idoneidad.

La mayoría de los criterios tienen asignadas una escala con dos valores lingüísticos: *"no soportado"* o *"soportado"*, indicando si la ontología tiene esa cualidad o no. Por ejemplo, un criterio con esta escala asignada es la característica: "el lenguaje de implementación soporta la definición de relaciones inversas". Otras, en cambio, permiten recoger la apreciación que de ese criterio tiene el usuario, tras haber hecho un análisis de sus propiedades. Para ello, se definen escalas que forman un conjunto equilibrado de valores lingüísticos, como por ejemplo: *"muy bajo"*, *"bajo"*, *"medio"*, *"alto"* y *"muy alto"*, que expresan cómo se ajusta el candidato en ese criterio a las necesidades del proyecto. Un ejemplo de criterio con esta escala lingüística sería: "las relaciones existentes en la ontología satisfacen las relaciones necesitadas en el proyecto". Para valorar este criterio concreto el usuario deberá, en primer lugar, analizar las relaciones que van a necesitarse en el proyecto, y luego, tras examinar las relaciones definidas en la ontología, determinar en qué grado cubren estas necesidades.

Al igual que propone la metodología *IDEAL* [Gom97], los valores lingüísticos son asociados a **intervalos difusos**, determinados por puntos de ruptura o puntos angulares, donde se puede definir una función de pertenencia, que indicará en qué grado se ajusta un valor concreto a uno de estos intervalos; teniendo en cuenta que, cuanto más se acerca la función a 1, más cierto es el

valor lingüístico, y cuanto más se acerca a 0, menos lo es. En la parte derecha de la figura 4.13 se muestra un ejemplo de la representación gráfica de los valores "medio" y "alto":

Como se aprecia en la parte izquierda de la figura 4.13, a cada valoración lingüística se le ha asignado un intervalo difuso con valores representativos en una escala de 0 a 10. De esta forma, se pueden realizar operaciones matemáticas básicas para intervalos, mediante cálculos en paralelo entre los puntos que ocupan las mismas posiciones en los intervalos angulares. Así, se pueden definir las operaciones de suma de intervalos (1) y producto por un número (2):

$$(a_1, a_2, a_3, a_4) + (b_1, b_2, b_3, b_4) = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4) \quad (1)$$

$$n * (a_1, a_2, a_3, a_4) = (n * a_1, n * a_2, n * a_3, n * a_4) \quad (2)$$

4.7.1.3.2. Asignación de Valores a las Criterios del AMCP a partir de las Instancias de la *Reference Ontology*

Para cada una de las ontologías candidatas, los criterios situados en los nodos hoja del AMCP, son valorados a partir de las instancias almacenadas en la RO. Las características de las dimensiones: "lenguajes de implementación", "metodologías de desarrollo", y "entornos software" son rellenadas directamente (y de forma automática con la aplicación ROIS) con la información de las instancias sobre estas dimensiones que se encuentra en la RO, y que aparecen en las tablas 4.19, 4.20, 4.21, 4.22, 4.23, 4.24, 4.25 y 4.26 de la sección 4.5.2.

Las características de las dimensiones "contenido" y "costes" situadas en los nodos hoja del AMCP son valoradas de forma distinta. En estos casos, el usuario asignará un valor de adecuación a partir de la información almacenada en la RO y que es proporcionada al usuario tras hacer consultas con la aplicación ROIS. Con la información mostrada, el usuario puede asignar un valor a estas características del "contenido" y "costes" de las ontologías candidatas respecto a las necesidades de su proyecto. Es decir, ROIS muestra toda la información sobre los conceptos, relaciones, taxonomía y axiomas que contienen las ontologías candidatas y, de esta forma, el usuario puede valorar cómo se ajusta ésta a la dimensión "contenido" de las ontologías según las necesidades de su proyecto. De igual forma, muestra la información correspondiente sobre los "costes" para poder completar la valoración de idoneidad de los factores correspondientes.

En la tabla 4.7 se indicaron las características sobre los conceptos almacenados en la RO relacionados con la dimensión "contenido", y en la tabla 4.19 se mostraron las relaciones entre conceptos que son almacenadas en la RO. Con esta información (sobre las características de los conceptos de la dimensión "contenido" de la RO y las relaciones entre ellos), el usuario puede

completar la valoración de las características usadas para la medición de idoneidad de la dimensión "contenido" del AMCP, que es la que aparece en la tabla 4.32. En la columna de la derecha de esta tabla se indican qué características y qué relaciones debe examinar el usuario para asignar un valor de idoneidad a las características de la dimensión "contenido" correspondientes.

De igual forma, para valorar la dimensión "costes" del AMCP, el usuario deberá analizar la información almacenada en la RO sobre las características que contienen esta información. En concreto, debe examinar la característica CO33 de la dimensión "contenido" (tabla 4.7), y las características T05, T07 y T08 de la dimensión "entornos" (tabla 4.5). La relación entre las características de la dimensión "costes" del marco multinivel de características y las características almacenadas en la RO se muestra en la tabla 4.33.

Código	Característica	Valorada desde:
	CONCEPTS	
C11	The essential concepts for the system are in the ontology	CC1, CC2, CC3
C12	The essential concepts for the system are in the superior levels of the ontology	CC1, CC2, CC3, R3
C13	The concepts are properly described in natural language	CC1, CC2
C14	The formal specification of each concept coincides with the description of each concept in natural language	CC1, CC2, CC3
C15	The attributes of the concepts accurately describe these concepts	CC1, CC2, CA1, CA2, CA3, CA4
C16	The number of concepts represented in the ontology is appropriate for the system	CO19
	RELATIONS	
C21	The essential relations for the system are defined in the ontology	CR1, CR2, CR3, CR6
C22	The relations between concepts are those it needed for the system	CR1, CR2, CR3, CR5, CR6, R9
C23	The formal specification of each relation coincides with the description of each relation in natural language	CR1, CR2, CR3, CR4, CR5, CR6
C24	Arity specified in relations is appropriate for the system	CR1, CR4, CC1, R9
C25	Formal properties of relations, such as reflexive, irreflexive, symmetrical, asymmetrical, antisymmetrical, transitive and intransitive, are specified in relations appropriately for the system	CR1, CR5
C26	The number of relations defined in the ontology is appropriate for the system	CO22, CO23
	TAXONOMY	
C31	The concepts of the ontology are classified from several perspectives appropriately for the system	CO24
C32	There are "Not-Subclass-Of " relations which are needed for the system	CO26
C33	There are exhaustive-partition relations which are needed for the system	CO27
C34	There are disjoint-partition relations which are needed for the system	C028
C35	The maximum depth in the inheritance of concepts is appropriate for the system	Calculado desde CC1, R3
C36	The average of subclasses by concepts is appropriate for the system	Calculado desde CC1, R3
	AXIOMS	
C41	There are defined axioms to solve queries	CNA1, CNA2, CNA3, CO30
C42	There are defined axioms to infer knowledge about attributes of concepts and instances of relations	CNA1, CNA2, CNA3, CO31
C43	There are defined axioms used to verify the consistency of terms in the ontology.	CNA1, CNA2, CNA3, CO32
C44	There are defined axioms as independent elements (those not linked to concepts)	CNA1, CNA2, CNA3, CO29
C45	The number of defined axioms in the ontology is appropriate for the system	CO29

Tabla 4.32. Características que deben examinarse de la RO para valorar la idoneidad de la dimensión "contenido" del marco multinivel de características.

	DIMENSIÓN COSTES	Valorada desde:
Cost1	The acquisition price of the ontology to be used in the system	CO33
Cost2	The costs to acquire the hardware and software resources in order to use the ontology	TO5
Cost3	The acquisition price of the access interfaces of the ontology	TO8
Cost4	The acquisition price of the edition and visualization tools of the ontology	TO7

Tabla 4.33. Características que deben examinarse de la RO para valorar la idoneidad de la dimensión "costes" del marco multinivel de características.

4.7.1.4. Paso 5: Cálculo y Comparación de los Valores de Idoneidad para las Ontologías Candidatas

A partir de los pesos del AMCP y de las valoraciones en los nodos hoja para cada ontología candidata, se obtienen recursivamente las valoraciones de los nodos situados en niveles superiores del AMCP. Finalmente, los valores obtenidos en el nodo raíz se identificarán con la idoneidad de las ontologías respecto al AMCP establecido. Estos últimos valores servirán para que el usuario pueda tomar la decisión de usar unas ontologías u otras.

Los cálculos para obtener los valores de los nodos superiores en el AMCP se harán mediante la fórmula ($S_n\ w_i\ v_i$), donde:

- w_i es el peso de importancia que posee ese nodo
- v_i es el intervalo difuso que corresponde a la valoración lingüística que tiene la ontología para los nodos hijo.

De esta forma, se hace la suma ponderada de todos los nodos hermanos que determinan el valor de idoneidad del nodo padre. Como ya se ha indicado, este cálculo se realizará desde los nodos inferiores de AMCP que tendrán asignado un valor directo (mediante el proceso explicado en el apartado 4.7.1.2.2), y se irá propagando recursivamente hacia arriba hasta obtener la valoración del nodo raíz, que se identifica con la valoración de idoneidad de la ontología candidata.

En la tabla 4.34, siguiendo el mismo ejemplo del factor "usabilidad de la metodología de desarrollo" que se mostró en la figura 4.12, se exponen valores de idoneidad obtenidos para dos supuestas ontologías O_1 y O_2.

Usabilidad	O_1		O_2	
	v_i	$w_i\ v_i$	v_i	$w_i\ v_i$
C1: (w=*0.3420*)	Alto(5.6, 6.6, 7.8, 8.8)	(1.9, 2.2, 2.6, 3.0)	Regular(3.4, 4.4, 5.6, 6.6)	(1.1, 1.5, 1.9, 2.2)
C2: (w=*0.5241*)	Regular(3.4, 4.4, 5.6, 6.6)	(1.7, 2.3, 2.9, 3.4)	Bajo(1.2, 2.2, 3.4, 4.4)	(0.6, 1.1, 1.5, 2.3)
C3: (w=*0.1339*)	MuyBajo(0, 0, 1.2, 2.2)	(0, 0, 0.1, 0.2)	Alto(5.6, 6.6, 7.8, 8.8)	(0.7, 0.8, 1, 1.1)
	Usabilidad de O_1 $\Sigma_n w_i v_i$:(3.6, 4.5, 5.6, 6.6)		**Usabilidad de O_2 $\Sigma_n w_i v_i$:(2.4, 3.4, 4.4, 5.6)**	

Tabla 4.34. Ejemplo de valoración de idoneidad del factor "Usabilidad" para dos ontologías candidatas.

Se pueden comparar los valores obtenidos en un criterio concreto, para varias ontologías candidatas, de forma parcial. De esta forma, el estudio puede centrarse en un aspecto concreto. En el ejemplo, los resultados de "usabilidad de la metodología", son representados gráficamente en la figura 4.14, donde se puede observar los valores de idoneidad frente a la escala lingüística que aparece en la parte izquierda de la figura. Se aprecia cómo la ontología O_1, para este

criterio, toma un valor algo mayor que "medio"; en cambio, la ontología O_2 toma un valor entre "medio" y "bajo".

Muy baja	0	0	1.2	2.2
Baja	1.2	2.2	3.4	4.4
Media	3.4	4.4	5.6	6.6
Alta	5.6	6.6	7.8	8.8
Muy alta	7.8	8.8	10	10

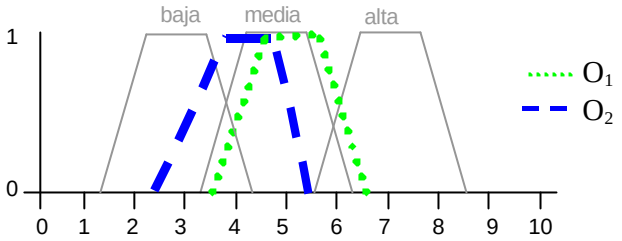


Figura 4.14. Resultado parcial de idoneidad de dos ontologías en el factor "Usabilidad de la metodología".

Como se ha dicho, los resultados calculados para los nodos que poseen nodos hijos deben propagarse hacia arriba para obtener progresivamente los resultados parciales de criterios más generales, hasta llegar al nodo raíz. En el ejemplo de la figura 4.15 se representan unos valores que pueden tomar una ontología O_1 en los factores "precisión", "usabilidad" y "madurez" de la metodología que serán utilizados, mediante la suma ponderada, para obtener la idoneidad de la dimensión "metodología".

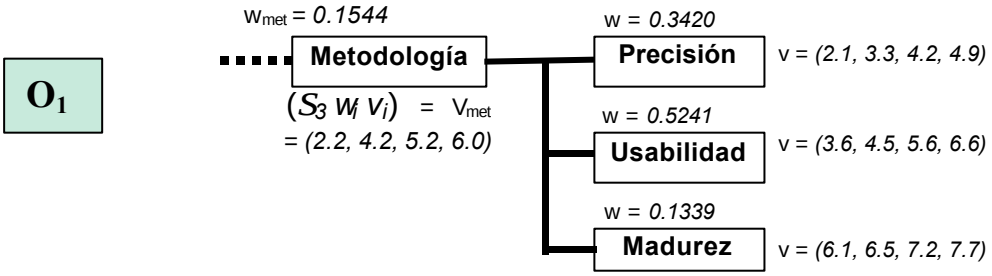


Figura 4.15. Resultado parcial de idoneidad de la dimensión *"Metodología"* para una ontología de ejemplo.

En proyectos de gran importancia, donde se pueda requerir un equipo de usuarios (o analistas), cada uno de ellos puede proporcionar sus propias valoraciones, y será necesario confrontarlas. En este escenario, todos los pasos donde haya discrepancias de opinión deben establecerse por consenso, utilizando técnicas similares a la *Wideband Delphi Technique* de Boehm [Boe81].

Tras realizar esta fase, el usuario debe elaborar un **informe de trabajo** que contemple los criterios analizados, la importancia que se le ha dado a cada uno de ellos, las ontologías estudiadas y los valores que las ontologías candidatas han tomado en cada uno de los criterios, concluyendo, mediante análisis comparativos, la decisión tomada. De esta forma podrá justificar, ante los directivos de su empresa, la elección de escoger unas ontologías frente a otras para usarlas en el nuevo proyecto.

4.7.1.5. Diferencias de *OntoMetric* con el Método de las Jerarquías Analíticas

A pesar de que *OntoMetric* sigue casi los mismos pasos y técnicas de cálculo en los procesos de valoración que AHP, se resumen aquí las diferencias:

- El usuario dispone de un conjunto de criterios ya definidos como base para realizar su juicio de valoración. Mediante el marco multinivel de características mostrado en la sección 4.4, organizado en forma de AMC, se ofrece al usuario los criterios fundamentales que deben ser analizados para decidir usar una ontología. A partir de este AMC por defecto, el usuario puede ajustarlo incorporando o eliminando características, y profundizando o podando el AMC, para abarcar las directrices y objetivos de su proyecto.
- Se ofrece una ponderación por defecto de cada uno de los criterios del AMC. Aunque esta ponderación no será, en la mayoría de los casos, la más indicada para todos los proyectos, sí sirve como base de comparación sobre el que se ajustarán los pesos de importancia del nuevo proyecto.
- Las diferentes alternativas (ontologías candidatas) cumplen unos requisitos obligatorios, ya que provienen del proceso de selección de ontologías (definido en la sección 4.6).
- Los valores de las características situadas en los nodos hoja del AMCP toman sus valores desde una escala de valoración fijada para cada una de las características. De esta forma se asegura que las valoraciones en esa característica, para varias ontologías candidatas, sean comparables. Las escalas asignadas a cada una de las características son de tipo lingüístico, de forma que al usuario le resulta más intuitivo hacer estas apreciaciones.
- Las escalas lingüísticas son identificadas con intervalos difusos, y los resultados de valoración obtenidos (final y parciales) son expresados también mediante intervalos difusos. La representación mediante gráficas, usando estos intervalos, para las ontologías candidatas son fácilmente comparables.

4.7.2. Herramienta Software para Asistir a la Aplicación del Método: *OntoMetric Tool*

Como se ha podido observar, en varias de las fases del método *OntoMetric* se requieren laboriosos procesos de recopilación de información de ontologías disponibles, cálculo para asignar los pesos de importancia en el AMCP, y cálculo para hallar las valoraciones de idoneidad de todos los nodos del árbol. Además, los valores obtenidos (final y parciales) son expresados mediante intervalos difusos que, para hacer cálculos y representarlos gráficamente, requieren un esfuerzo adicional.

Realizar estas tareas de forma "manual" requeriría un enorme esfuerzo, y con bastante probabilidad haría que el potencial usuario rechazara el uso del método, por el notable consumo de tiempo que conlleva. Por estas razones se ha desarrollado una aplicación software (OT) que asiste en todos los pasos del método. Esta herramienta realiza las siguientes tareas:

- Permite usar el AMCP por defecto, con los pesos obtenidos desde las encuestas de opinión de los usuarios, o puede utilizarse otro AMCP existente que hubiera sido creado para una valoración anterior de otro proyecto.
- Importa la información sobre las características de las ontologías candidatas seleccionadas desde la aplicación ROIS. De esta forma, automáticamente se completan los valores que toman las características de las dimensiones de "lenguajes de implementación", "metodologías de desarrollo" y "entornos software ", situadas en los nodos hoja del AMC. Los valores de las características de las dimensiones "contenido" y "costes" deben ser completados por los usuarios a partir de la información proporcionada por ROIS. Estos valores pueden indicarse de forma intuitiva, seleccionando el valor correspondiente en la escala lingüística asignada a cada nodo.
- Permite ajustar el AMCP, pudiendo ampliar o podar las ramas según las necesidades de evaluación del proyecto en estudio.
- De forma gráfica, permite rellenar las matrices de comparación de pares para indicar la importancia de un criterio frente a los otros.
- A partir de las matrices de comparación de pares, calcula los pesos de importancia de criterios (los autovectores) y se los asigna a los nodos correspondientes del AMCP.
- Realiza los cálculos de los valores de idoneidad de cada uno de los criterios, y del valor del nodo raíz que mide la idoneidad de cada una de las ontologías candidatas.
- Seleccionando un criterio concreto (incluyendo el raíz), construye gráficas comparativas que representan los intervalos difusos que toman las ontologías candidatas en ese criterio. Las gráficas indican los valores de cada ontología en cada criterio, mostrando como base de comparación la escala lingüística asignada.
- Realiza informes impresos donde se muestra la información sobre las características de las ontologías candidatas, la importancia que se ha dado a cada criterio, los valores parciales que se han obtenido para cada criterio, e incluye gráficas comparativas que ayudarán al usuario a decidir la elección de unas ontologías frente a otras.

Capítulo 5

Evaluación de OntoMetric con Casos de Estudio

5.1. INTRODUCCIÓN

En este capítulo se muestran varios casos de prueba del método *OntoMetric*, utilizando como soporte tecnológico las aplicaciones *Reference Ontology Instances Selector* (ROIS) y *OntoMetric Tool* (OT). ROIS es usada en el proceso de selección de ontologías candidatas, de lenguajes de implementación de ontologías, de metodologías de desarrollo de ontologías y de entornos software de construcción de ontologías. OT sirve para, tomando los valores de las instancias obtenidos a partir de la aplicación ROIS, obtener la idoneidad de las ontologías candidatas, de lenguajes de implementación de ontologías, de metodologías de desarrollo de ontologías y de entornos software de construcción de ontologías, mediante el uso del método de las jerarquías analíticas adaptado con valores lingüísticos.

Se han realizado dos tipos de pruebas:

- **Pruebas individuales**, por separado, sobre cada una de las dimensiones (contenidos, lenguajes, metodologías y entornos) que tienen instancias en el marco multinivel de características. La dimensión "costes de uso" no puede consultarse como entidad independiente, ya que deben examinarse características de las ontologías existentes y de los entornos software relacionados con esas ontologías. Ejemplos sobre los tipos de pruebas individuales realizadas son: seleccionar un lenguaje que permita crear atributos globales y que sea de marcado *web*; O seleccionar un entorno software que permita visualizar de forma gráfica la taxonomía de conceptos, que avise a los usuarios del mismo grupo cuando se ha modificado una ontología, y que permita exportar la ontología a lenguajes de marcado *web*.
- **Pruebas conjuntas**, que combinan aspectos de varias dimensiones. Es decir, se van a señalar unos valores en una serie de características de varias de las dimensiones, y las ontologías que se hallen deben cumplir las exigencias indicadas en todas las dimensiones. Por ejemplo: seleccionar ontologías en el dominio de "componentes electrónicos", implementadas en un lenguaje que permita herencia múltiple, que las licencias de uso de la ontología sean gratis, y que algún entorno software que pueda editar la ontología posea verificación de la consistencia en los procesos de integración de ontologías.

Se han realizado 15 casos de pruebas individuales para cada una de las dimensiones: contenido, lenguajes, metodologías, y entornos, y 15 casos de pruebas conjuntas. Con estas 75 pruebas realizadas se evalúa la aplicabilidad del método en cada dimensión de forma independiente y, posteriormente, considerando todas las dimensiones. En cada uno de los casos de prueba presentados en este capítulo se incluyen los siguientes apartados:

A. **Selección de las características que van a formar la consulta** en las pruebas individuales y en las pruebas conjuntas. Una vez seleccionadas las características, se le asignan valores específicos y valores mínimos. Tales valores dependerán de las directrices establecidas en el proyecto. ROIS, como muestra la figura 5.1, permite construir consultas simples y compuestas sobre características, relacionadas con el contenido, lenguajes, metodologías, entornos y costes. Los resultados de la búsqueda serán las instancias de ontologías, lenguajes, metodologías y entornos, que serán los considerados como candidatos.

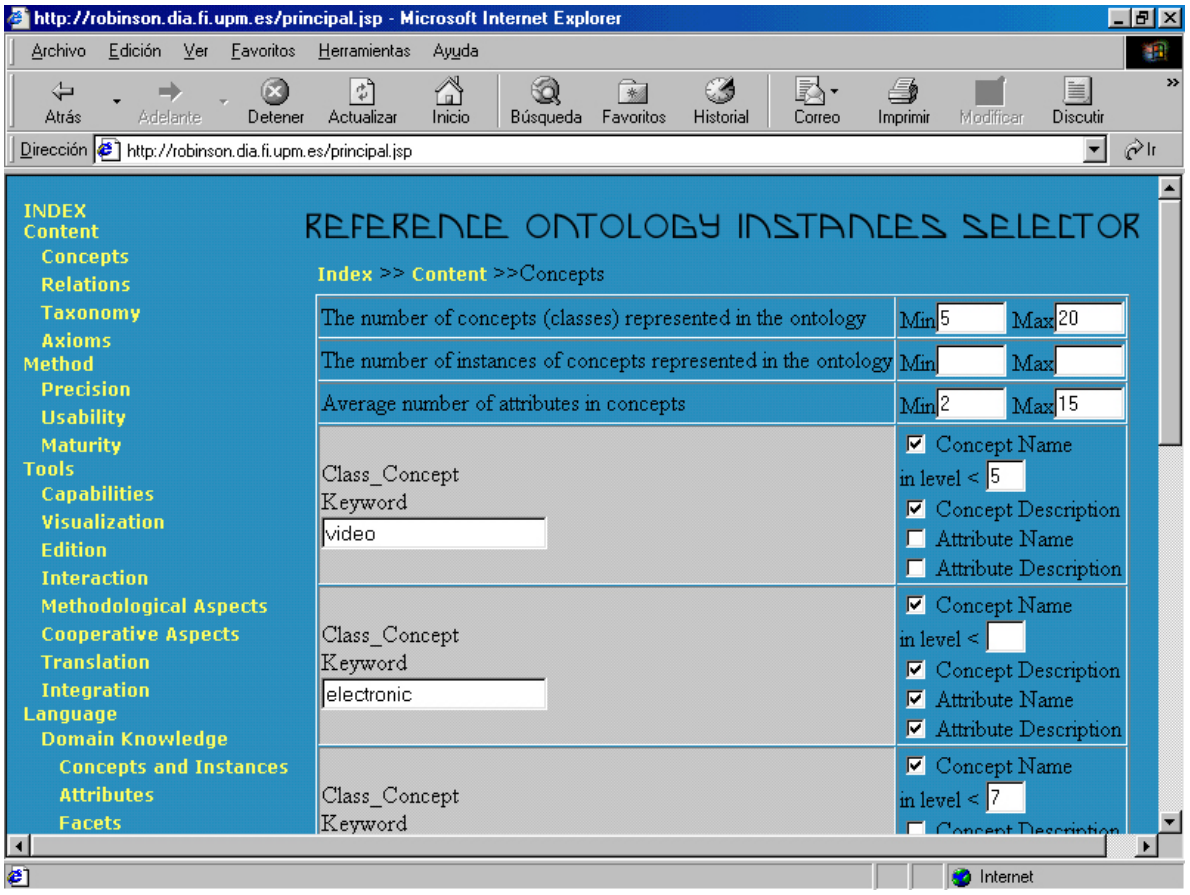


Figura 5.1. Ejemplo de creación de una consulta en la aplicación ROIS.

B. **Identificación "manual" de los elementos candidatos**. Se hace un análisis "manual" de las características indicadas en los casos de prueba para determinar qué instancias de la RO cumplen esos requisitos. Esta fase no es necesaria realizarla en una medición real de un proyecto ya que ROIS lo hace de forma automática; aparece en estos casos de prueba

para comprobar que ROIS realiza la selección de candidatos adecuadamente.

C. Tras ejecutar la aplicación ROIS, se obtienen **los elementos candidatos**. Es decir, el conjunto de instancias de ontologías, lenguajes, metodologías y entornos que satisfacen los valores indicados. Se comprueba la similitud con los resultados obtenidos manualmente en el paso 2. Si son iguales significa que ROIS funciona sin errores.

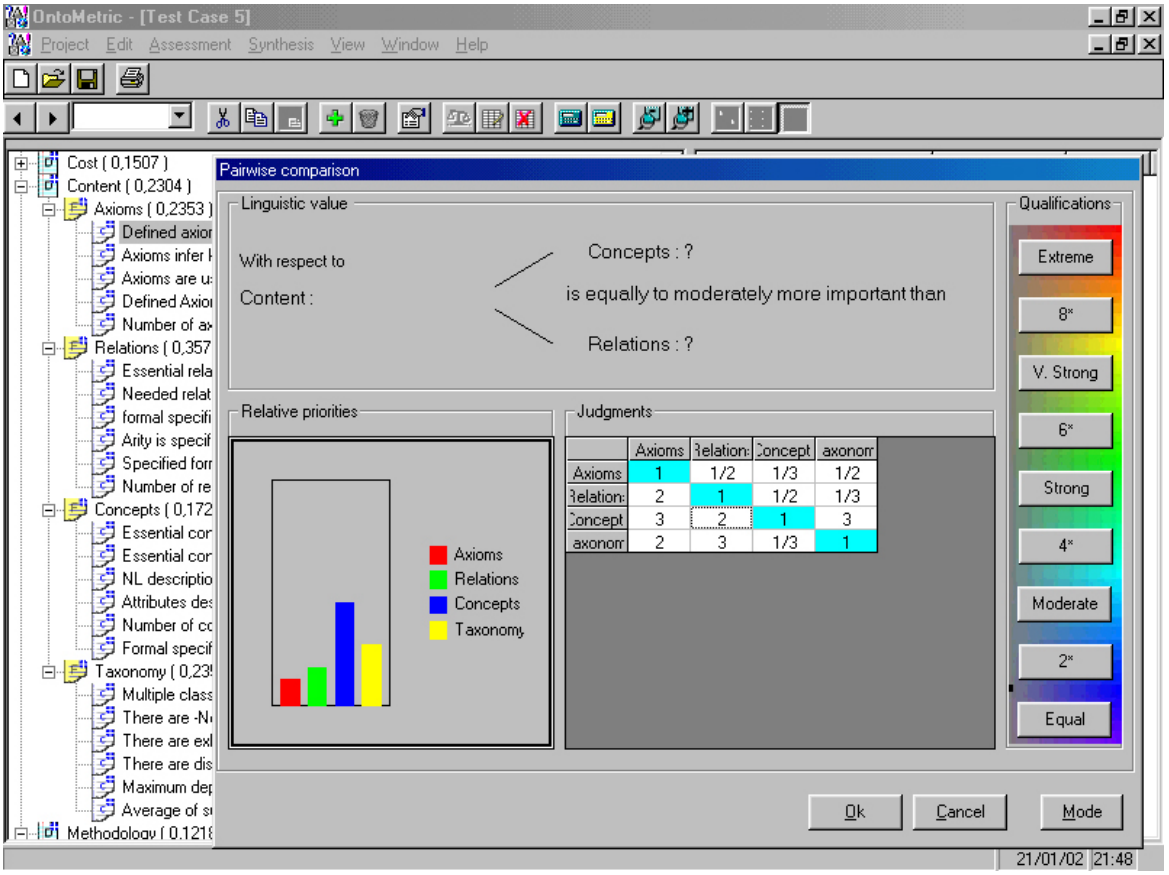


Figura 5.2. Ejemplo de asignación de pesos mediante la matriz de comparación de nodos hermanos, implementado en OT.

D. Para cada uno de los criterios (dimensiones, factores, características y subcaracterísticas) que intervienen en las dimensiones sobre las que se hace la consulta, se asignan **pesos de importancia**. Para indicar tales pesos se han utilizado, en algunos casos, los pesos por defecto mostrados en la sección 4.7.1.2.2. Para otros casos, se han calculado los pesos de importancia mediante el proceso de asignación de pesos a partir de las matrices de comparación de pares, implementado en OT, que se describe en la sección 4.7.1.2.1. Como se muestra en la figura 5.2, OT tiene implementado, de forma gráfica, un módulo muy intuitivo para que el usuario indique estos valores y, a partir de estas matrices, realizar los cálculos de los pesos de forma automática, obteniendo los pesos para cada conjunto de criterios hermanos.

E. Para todas las características de la dimensión sobre la que se está haciendo la consulta, se muestran los **valores de las características en cada instancia candidata encontrada**.

Los valores de las características de las dimensiones de lenguajes de implementación, metodologías de desarrollo y entornos software se rellenan automáticamente usando ROIS. Los valores de las características de las dimensiones "contenido" y "costes" se completan por el usuario a partir de la información sobre "ontologías" y "entornos" que muestra la aplicación ROIS, a partir de las instancias que tiene almacenadas, de la forma que se explicó en el apartado 4.7.1.3.2.

F. Se calcula la **medida de idoneidad para cada una de los elementos candidatos**. La aplicación OT, con los valores obtenidos en los puntos 4 y 5, realiza el cálculo de idoneidad de cada uno de los elementos candidatos, obteniendo un valor identificado con un intervalo difuso. Estos resultados se pueden comparar gráficamente de forma global contrastando los resultados entre las instancias candidatas, como el ejemplo que aparece en el caso de prueba 5 en la gráfica 5.11; o de forma individual contrastando cada una de las instancias obtenidas, con los valores de una determinada escala de valoración, como el ejemplo que aparece en el caso de prueba 5 en la gráfica 5.12.

5.2. CASOS DE PRUEBA PARA EVALUAR CADA DIMENSIÓN INDIVIDUALMENTE

El objetivo principal de este trabajo es ofrecer un método que indique la idoneidad de una o varias ontologías para un proyecto determinado; de esta forma, se pueden realizar comparaciones de idoneidad cuando existan varias ontologías candidatas. Además, el método permite la **valoración parcial de idoneidad** de cada una de las dimensiones, factores y características no situadas en nodos hoja (especificadas con otras características), que deben considerarse para elegir una ontología frente a otras. Esto puede realizarse debido a la estructura multinivel que ofrece el árbol multinivel de características. La herramienta software *OntoMetric Tool* realiza cálculos y gráficas parciales de valoración, de forma que un usuario puede realizar un estudio más detallado de un criterio determinado, para precisar o justificar su decisión.

En los siguientes apartados se ofrecen algunos de los resultados obtenidos, a modo de ejemplo, de casos de prueba sobre valoraciones parciales en las dimensiones "lenguajes de implementación", "metodologías de desarrollo", "entornos software" y sobre el "contenido" de las ontologías. No se ha considerado oportuno mostrar un caso de prueba sobre los "costes de uso" de ontologías ya que, actualmente, los valores en las características que llevan a determinar esta dimensión no son significativamente comparables, debido a que ninguna de las ontologías conocidas, excepto *CYC* [Len90], tienen asignado un precio, y muy pocos entornos software requieren licencias de uso y, si la tienen, no está indicado en la *web*.

5.2.1. Casos de Prueba para la Dimensión "Lenguajes de Implementación de Ontologías"

Las instancias recogidas en la ontología RO en el concepto "lenguaje de implementación" son: *Ontolingua*, *CycL*, *LOOM*, *FLogic*, *OCML*, *OKBC*, *XOL*, *SHOE*, *OIL*, *OML/CKML*, *DAML+OIL* y *RDF(S)*. El conjunto de características que se identificaron en el apartado 4.4.1.4 para la dimensión "lenguajes de implementación" de la RO se muestran en las tablas 4.2 y 4.3. Los valores que toman estas instancias para las características relativas a los factores "conocimiento del dominio" y "mecanismos de inferencia" aparecen en las tablas 4.20 a la 4.23 de la sección 4.5.2 para los lenguajes clásicos y de marcado *web*.

A continuación se expone un caso de prueba que muestra el resultado de usar la aplicación ROIS para la búsqueda de lenguajes candidatos, y OT para evaluar la idoneidad de estos lenguajes en un proyecto determinado. En la tabla 5.1 se indican los valores exigidos a las características de los lenguajes candidatos. Los pesos indicados en la tabla 5.2 se han obtenido mediante el proceso de cálculo desde las matrices de comparación de pares, implementado en OT. Es decir, se ha asignado la importancia que tiene cada criterio (características y factores en cada caso) con sus criterios afines, y la aplicación OT ha calculado los pesos que aparecen en la tabla 5.2.

CASO DE ESTUDIO 1 (ELECCIÓN DE LENGUAJES):

A. SELECCIÓN DE CARACTERÍSTICAS Y ASIGNACIÓN DE VALORES

Código	Característica	Valor de búsqueda
LO4	Quality of manuals	*Medium*
L111	It allows us to create instances of class	*Supported*
L121	Class Attributes (own slots)	*Supported*
L123	Local Attributes	*Supported*
L132	The facet "Type of value" (or another similar one) exists	*Supported*
L143	The language allows us to define ad-hoc relations	*Supported*
L144	The language allows us to constrain the type of arguments in relations	*Supported*
L153	The language allows us to define exhaustive partitions	*Supported*
L211	The language allows multiple inheritance	*Supported*
L225	You can build a new inference engine	*Supported*

Tabla 5.1. Valores exigidos a los lenguajes en el caso de prueba 1.

B. CANDIDATOS ESPERADOS

- Ontolingua
- LOOM
- DAML+OIL
- OCML
- FLogic

C. CANDIDATOS ENCONTRADOS POR *ROIS*

- Ontolingua
- LOOM
- DAML+OIL
- OCML
- FLogic

D. y E. PESOS Y VALORES DE CARACTERÍSTICAS EN LOS LENGUAJES CANDIDATOS

Código	Peso	**Instancia1**	**Instancia2**	**Instancia3**	**Instancia4**	**Instancia5**
Código	**Peso**	**Instancia1**	**Instancia2**	**Instancia3**	**Instancia4**	**Instancia5**
DESCRIPTIVE CHARACTERISTICS						
L01	--	Ontolingua	LOOM	DAML+OIL	OCML	Flogic
L02	--	Classical	Classical	Markup	Classical	Classical
L03	--	Frame-based, Classical_logic	Description _Logic	Frame-based, Description_Logic	Frame-based, Classical_logic	Frame-based, Classical_logic
L04	--	High	Medium	High	Medium	Medium
L05	--	Supported	Supported	Supported	Supported	Non_supported
L06	--	http://ontolingua.stanford.edu	http://www.isi.edu/isd/LOOM/LOOM-HOME.html	http://www.daml.org	http://kmi.open.ac.uk/projects/ocml/	--
L07	--	Stanford University Knowledge Systems Laboratory	Southern California ISI	DAML project	VITAL project	--
L08	--	6.0	4.0	0.6	--	--
L09	--	1997-11-01	2000-05-01	2001-12-5	2001-01-01	1995-01-01
DOMAIN KNOWLEDGE (FACTOR)	0.54					
CONCEPTS/ INSTANCES/ FACTS/ CLAIMS	0.1699					
L111	0.3128	Supported	Supported	Supported	Supported	Supported
L112	0.2548	Supported	Supported	Supported	Supported	Supported
L113	0.1599	Supported	Non_supported	Supported	Supported	Non_supported
L114	0.1831	Supported	Supported	Non_supported	Supported	Supported
L115	0.0894	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
ATTRIBUTES	0.1583					
L121	0.18	Supported	Supported	Supported	Supported	Supported
L122	0.222	Supported	Supported	Supported	Supported	Supported
L123	0.1501	Supported	Supported	Supported	Supported	Supported
L124	0.1294	Supported	Supported	Supported	Non_supported	Non_supported
L125	0.1526	Supported	Supported	Supported	Supported	Non_supported
L126	0.1656	Non_supported	Non_supported	Non_supported	Non_supported	Supported
FACETS	0.1202					
L131	0.1852	Supported	Supported	Non_supported	Supported	Supported
L132	0.229	Supported	Supported	Supported	Supported	Supported
L133	0.2279	Supported	Supported	Supported	Supported	Supported
L134	0.1743	Non_supported	Supported	Non_supported	Supported	Non_supported
L135	0.1831	Supported	Supported	Non_supported	Non_supported	Non_supported
RELATIONS	0.1466					
L141	0.1106	Supported	Supported	Non_supported	Supported	Supported
L142	0.131	Supported	Supported	Non_supported	Supported	Non_supported
L143	0.2092	Supported	Supported	Supported	Supported	Supported
L144	0.1322	Supported	Supported	Supported	Supported	Supported
L145	0.1661	Supported	Supported	Non_supported	Supported	Non_supported
L146	0.1435	Non_supported	Supported	Non_supported	Supported	Supported
L147	0.1075	Supported	Non_supported	Supported	Non_supported	Non_supported
TAXONOMY	0.1497					
L151	0.1978	Supported	Supported	Supported	Supported	Supported
L152	0.1108	Supported	Supported	Non_supported	Non_supported	Non_supported
L153	0.1585	Supported	Supported	Supported	Supported	Supported
L154	0.1545	Supported	Supported	Supported	Supported	Supported
L155	0.194	Supported	Supported	Supported	Supported	Supported
L156	0.1845	Non_supported	Supported	Supported	Non_supported	Unknown
AXIOMS	0.1311					
L161	0.3016	Supported	Supported	Non_supported	Supported	Non_supported
L162	0.1697	Supported	Non_supported	Non_supported	Supported	Non_supported
L163	0.2668	Supported	Supported	Non_supported	Supported	Supported
L164	0.2619	Supported	Non_supported	Non_supported	Non_supported	Non_supported
PRODUCTION RULES	0.1243					
L171	0.1643	Non_supported	Supported	Non_supported	Supported	Non_supported
L172	0.1113	Non_supported	Supported	Non_supported	Supported	Non_supported
L173	0.1621	Non_supported	Non_supported	Non_supported	Supported	Non_supported
L174	0.1885	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L175	0.1857	Non_supported	Supported	Non_supported	Supported	Non_supported
L176	0.1881	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported

Tabla 5.2. Pesos asignados y valores en las instancias candidatas para el caso de prueba 1 sobre lenguajes de implementación de ontologías **(1/2)**.

INFERENCE MECHANISM (FACTOR)	0.45	Ontolingua	LOOM	DAML+OIL	OCML	FLogic
REASONING POTENTIAL	0.525					
L211	0.1744	Supported	Supported	Supported	Supported	Supported
L212	0.1426	Supported	Supported	Non_supported	Supported	Supported
L213	0.1213	Non_supported	Supported	Non_supported	Non_supported	Supported
L214	0.1606	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L215	0.1684	Supported	Supported	Non_supported	Supported	Supported
L216	0.1508	Supported	Supported	Non_supported	Supported	Supported
L217	0.082	Non_supported	Supported	Non_supported	Supported	Supported
INFERENCE ENGINE	0.475					
L221	0.2044	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L222	0.1625	Non_supported	Supported	Non_supported	Non_supported	Non_supported
L223	0.1936	Non_supported	Non_supported	Non_supported	Non_supported	Supported
L224	0.2424	Non_supported	Non_supported	Non_supported	Non_supported	Non_supported
L225	0.1972	Supported	Supported	Supported	Supported	Supported

Tabla 5.2. Pesos asignados y valores en las instancias candidatas para el caso de prueba 1 sobre lenguajes de implementación de ontologías **(2/2)**.

F. MEDIDAS ASIGNADAS POR *ONTOMETRIC* A LOS CANDIDATOS Y GRÁFICA DE COMPARACIÓN

Instancia	Intervalo obtenido
Ontolingua	(1.829, 3.659, 7.015, 8.486)
OCML	(2.134, 4.268, 7.634, 8.795)
LOOM	(2.291, 4.582, 7.951, 8.961)
Flogic	(2.216, 4.433, 7.801, 8.884)
DAML+OIL	(1.161, 2.323, 5.658, 7.797)

Tabla 5.3. Valores de idoneidad asignados a los lenguajes candidatos en el caso de prueba 1.

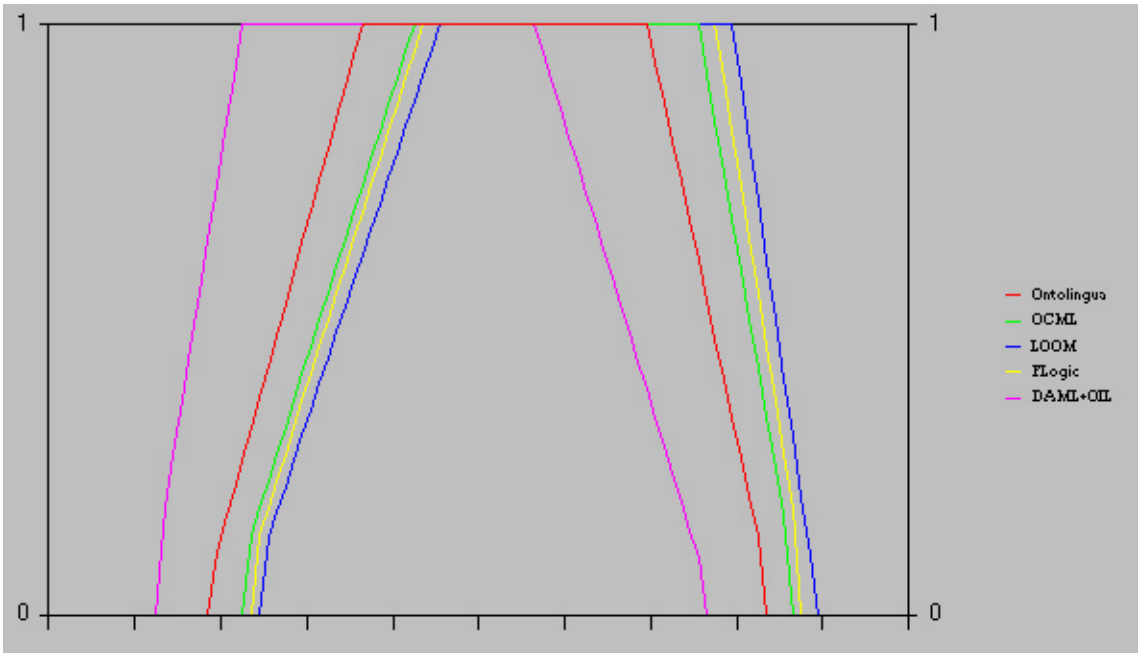


Figura 5.3. Gráfica de comparación mostrada por OT para el caso de estudio 1.

Lenguaje aconsejado: LOOM

Con las características indicadas en el punto 1, se han eliminado como candidatos los lenguajes *XOL*, *SHOE*, *OIL*, *OML/CKML* y *OKBC*. Si observamos las características en las que se ha

exigido un valor determinado, la característica L153, que indica la posibilidad de definir en el lenguaje particiones exhaustivas, ha sido la que ha discriminado a los lenguajes citados.

En los intervalos de idoneidad obtenidos mostrados en la tabla 5.3 y en la gráfica comparativa de la figura 5.3 se observa que el lenguaje que aconseja *OntoMetric* es *LOOM* por obtener una medida superior a los demás lenguajes candidatos, aunque con poca diferencia con respecto a los lenguajes *FLogic* y *OCML*. No se puede decir que exista una característica determinante en la que *LOOM* supere a los demás lenguajes, si no que, con los pesos asignados en este caso, *LOOM* ofrece un mejor valor global de idoneidad.

5.2.2. Casos de Prueba para la Dimensión "Metodologías de Desarrollo de Ontologías"

Las instancias recogidas en la ontología RO en el concepto "Metodología de Desarrollo" son: *CYC*, Uschold y King, Grüninger y Fox, la propuesta de Bernaras y colegas, la metodología de *SENSUS*, *METHONTOLOGY* y, *Co4 Methodology* y *(KA)^2 Methodology*. El conjunto de características identificadas para la dimensión "Metodología de Desarrollo" de la RO se muestran en la tabla 4.4 del apartado 4.4.1.5. Los valores que toman estas instancias para las características descriptivas y sobre los factores de "precisión", "usabilidad", y "madurez" se indicaron en las tablas 4.24 y 4.25 de la sección 4.5.2.

A continuación se expone un caso de prueba que muestra el resultado de usar la aplicación ROIS para la búsqueda de metodologías candidatas que cumplen los valores indicados en la tabla 5.4, y OT para evaluar la idoneidad de estas metodologías en un proyecto determinado. En este caso, los pesos indicados que se muestran en la tabla 5.5 se han obtenido mediante el proceso de cálculo desde las matrices de comparación de pares, implementado en OT.

CASO DE ESTUDIO 2 (ELECCIÓN DE METODOLOGÍAS):

A. SELECCIÓN DE CARACTERÍSTICAS Y ASIGNACIÓN DE VALORES

Código	Característica	Valor de búsqueda
M09	Activities at the life cycle	*(incluido) Requeriment process*
M11	Well defined the development phases	*High*
M21	Activities and techniques of each phase are easily understandable	*Medium*
M31	Number of ontologies developed for real projects following the methodology.	*Low*
M32	Number of different domains in those that the methodology has been used	*Low*
M33	Importance and number of significant projects in those that the methodology has been used	*High*

Tabla 5.4. Valores exigidos a las metodologías en el caso de prueba 2.

B. CANDIDATOS ESPERADOS

- Gruninger-Fox_methodology
- Bernaras_et_al_ methodology
- METHONTOLOGY
- SENSUS-based_methodology

C. CANDIDATOS ENCONTRADOS POR *ROIS*

- Gruninger-Fox_methodology
- Bernaras_et_al_ methodology
- METHONTOLOGY
- SENSUS-based_methodology

D. y E. PESOS Y VALORES DE CARACTERÍSTICAS EN LAS METODOLOGÍAS CANDIDATAS

Código	Peso	Instancia1	Instancia2	Instancia3	Instancia4
M01	--	Gruninger-Fox_methodology	Bernaras_et_al_ methodology	METHONTOLOGY	SENSUS-based_methodology
M02	--	It builds a logic model of knowledges by a formal description of requeriments, whose are obtained from ontology specifications. Then, this description is formalized.	This methodology builds ontologies by integrating and generalizing of knowledge bases.	It is enable the construction of ontologies at the knowledge level. It is inspired in IEEE Standard for Developing Sofware Life Cycle Process.	It is a methodology used to build ontologies starting from SENSUS abstract concepts.
M03	--	Low	High	High	Very_Low
M04	--	Capture of scenarios | Formulation of informal competency questions | Specification of terms with formal language | Formulation of formal competency questions | Formal specification of axioms and terms | Establishing completeness	Specification of the application | Preliminary design based in top-level categories | Ontology refinement and structuring	Specification | Conceptualization | Formalization | Implementation	Identification of seed terms | Linking seed terms to SENSUS |
M05	--	Classical_Logic	None	None	Semantic-networks
M06	--	Application-semidependent	Application-dependent	Application-independent	Application-semidependent
M07	--	Middle-out	Top-down	Middle-out	Down-Top
M08	--	Not defined totally	None	Evolving prototypes	None
M09	--	Requeriment process | Design process | Implementation process | Integral process (Part.)	Requeriment process | Design process | Implementation process	Project management processes (Partially) | Requeriment process | Design process | Implementation process | Post-development processes (Part.) | Integral process (Part.)	Requeriment process | Implementation process
M010	--	Unknown	Unknown	Some techniques missing	Unknown
M011	--	TOVE implementation tools	Unknown	ODE | WebODE	Ontosaurus
M012	--	TOVE ontology	Ontology for electrical networks	CHEMICALS | Environmental pollutans ontologies | Reference-Ontology | Restrutured (KA)2	SENSUS
PRECISION	**0,3275**				
M11	0,2672	Very_High	High	Very_High	High
M12	0,2026	Very_Low	Very_Low	Very_High	Very_Low
M13	0,1447	Very_Low	Very_Low	Low	Very_Low
M14	0,2132	Low	Very_Low	Very_High	Very_Low
M15	0,1723	Very_Low	Very_Low	Very_High	Very_Low
USABILITY	**0,4126**				
M21	0,3874	Medium	Medium	Very_High	Medium
M22	0,1692	Very_Low	Very_Low	High	Very_Low
M23	0,4434	Non_supported	Non_supported	Supported	Non_supported
MATURITY	**0,2594**				
M31	0,4126	Low	Low	High	Low
M32	0,3275	Low	Low	Very_High	Low
M33	0,2599	High	High	High	High

Tabla 5.5. Pesos asignados y valores en las instancias candidatas para el caso de prueba 2 sobre metodologías de desarrollo de ontologías.

F. MEDIDAS ASIGNADAS POR ONTOMETRIC A LOS CANDIDATOS Y GRÁFICA DE COMPARACIÓN

Instancia	Intervalo obtenido
Gruninger-Fox_methodology	(1.918, 2.458, 4.083, 5.413)
Bernaras_et_al_ methodology	(1.642, 2.149, 3.734, 5.154)
METHONTOLOGY	(5.774, 7.194, 8.797, 9.249)
SENSUS-based_methodology	(1.642, 2.149, 3.734, 5.154)

Tabla 5.6. Valores de idoneidad asignados a las metodologías candidatas en el caso de prueba 2.

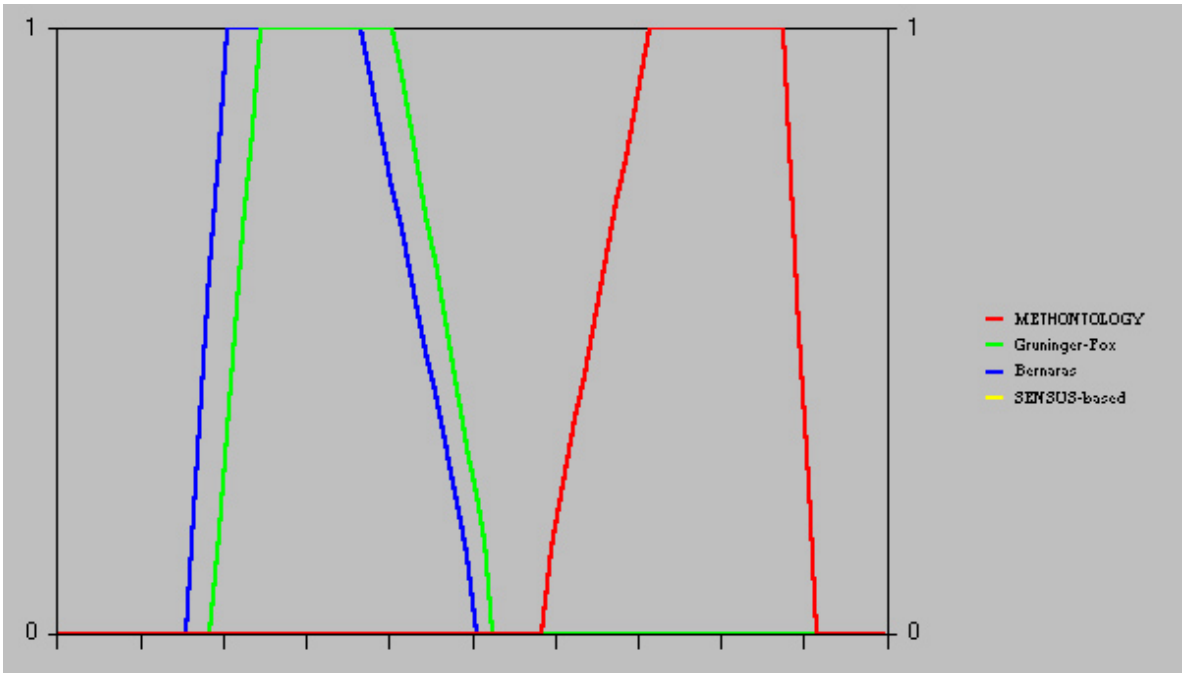


Figura 5.4. Gráfica de comparación mostrada por OT para el caso de estudio 2.

Metodología aconsejada: *METHONTOLOGY*

En este caso, cuatro metodologías han pasado el proceso de selección con los requisitos indicados a ROIS. La característica descriptiva M09 ha servido para seleccionar las metodologías que tuvieran una actividad en su ciclo de vida denominada "proceso de requisitos". En este caso fueron cinco metodologías (las cuatro que aparecen en la tabla 5.5 y la metodología de Uschold y King) las que superaron esta discriminación. Con la característica M11 se descartó esta última, ya que se ha exigido que tengan en la delimitación de las fases de desarrollo un valor "alto" o mayor, y la metodología de Uschold y King tiene este valor como "muy bajo".

En la tabla 5.6 y en la gráfica de la figura 5.4 se observa como *METHONTOLOGY*, con los pesos asignados, obtiene unos valores en la medida de idoneidad muy superiores a las metodologías comparadas. Si se observa la tabla 5.5, *METHONTOLOGY* tiene, comparativamente, valores superiores en la mayoría de las características usadas para valorar la idoneidad.

Respecto a la diferencia entre los demás candidatos, no se observa grandes diferencias en este caso, y en la gráfica no aparece la metodología *Sensus*, ya que su valor se encuentra superpuesto con el valor de la metodología de *Bernaras*, pues tienen idénticos valores en todas las características usadas para valorar la idoneidad, como se observa en la tabla 5.5.

5.2.3. Casos de Prueba para la Dimensión "Entornos Software de Desarrollo de Ontologías"

Las instancias recogidas en la ontología RO en el concepto "Entornos Software de Desarrollo" de ontologías son: *Ontolingua Server*, *Tadzebao-WebOnto*, *OntoEdit*, *WebODE*, *OILed*, y *Protégé2000*. Los valores de las características para estas instancias pueden verse en la tabla 4.26 de la sección 4.5.2. El conjunto de características identificadas para "Entornos Software" de la RO se muestra en la tabla 4.5 de la sección 4.4.1.6.

A continuación se expone un caso de prueba que muestra el resultado de usar la aplicación ROIS para la búsqueda de entornos software candidatos con los valores indicados en la tabla 5.7, y OT para evaluar la idoneidad de estos entornos en un proyecto determinado. Los pesos que aparecen en la tabla 5.8 se han obtenido mediante el proceso de cálculo desde las matrices de comparación de pares, implementado en OT.

CASO DE ESTUDIO 3 (SELECCIÓN DE ENTORNOS DE DESARROLLO):

A. SELECCIÓN DE CARACTERÍSTICAS Y ASIGNACIÓN DE VALORES

Código	Característica	Valor de búsqueda
T11	The tool can be installed locally.	*Supported*
T13	The tool works on the Internet with the use of a browser.	*Supported*
T14	Regarding the user interface clarity (i.e. the organization of options, their uniformity, absence of ambiguities in the options, contextual help, etc.).	*Low*
T15	The speed to visualize or to edit terms.	*Medium*
T22	It allows us to select the information with the level of detail required in each moment.	*Supported*
T23	It allows us to see the taxonomy of the concepts (not necessarily graphic).	*Supported*
T24	It allows us to see, graphically, the ad-hoc relations of the concepts	*Supported*
T31	With this tool, we can make the same representations of the knowledge that can be implemented directly with the specification language	*Supported*
T32	The tool allows us to modify the terms at any moment. That is to say, we can edit the ontology, add new terms, modify and erase them, etc.	*Supported*
T33	The tool allows us to modify the taxonomic relations in a graphic way.	*Supported*
T41	The ontology can be exported from the server and used locally	*Supported*
T52	The tool covers the most important activities (or those suitable to be assisted by software) of the life cycle of ontological development.	*Supported*
T54	The tool verifies the changes that have been carried out in the terms of the ontology. They check the consistency of the ontology.	*Supported*
T61	The tool allows us to create work groups to edit the same ontology.	*Supported*
T62	The tool allows us to perform synchronous edition of the same ontology.	*Supported*
T64	The tool locks the published concrete terms thus refusing modifications from other users.	*Supported*
T75	We might lose some semantics in a translation from one language to another. Some tools do this translation and lose a minimal amount of the semantics. This minimal loss is caused by the fact that the object language is less expressive than the source language	*Supported*

Tabla 5.7. Valores exigidos a los entornos en el caso de prueba 3.

B. CANDIDATOS ESPERADOS

- OilEd
- OntoEdit
- Protégé2000

C. CANDIDATOS ENCONTRADOS POR *ROIS*

- OilEd
- OntoEdit
- Protégé2000

D. y E. PESOS Y VALORES DE CARACTERÍSTICAS EN LOS ENTORNOS CANDIDATOS

Código	Peso	Instancia1	Instancia2	Instancia3
T01	--	OilEd	OntoEdit	Protégé2000
T02	--	University of Manchester, the Free University of Amsterdam and Interprice GmbH.	Knowledge Management Group of Karlsruhe University, Ontoprise	Stanford Medical Informatics at the Stanford University School of Medicine
T03	--	Java 1.2	Java 1.3	Java 1.3
T04	--	PC	PC	PC
T05	--	1000	1000	1000
T06	--	2-7-2001	3-7-2001	1-10-2000
T07	--	0	0	0
T08	--	0	0	0
T09	--	--	--	--
T010	--	DAML, RDFS	DAML, RDFS, F-Logic	F-Logic, OIL, Ontolingua, RDFS
T011	--	DAML, RDFS, SHIQ, Dotty	DAML, RDFS, F-Logic	F-Logic, OIL, Ontolingua, RDFS
T012	--	(KA)2 Ontology, The CIA World Factbook, 12 Ontologies from the Tourist Domain, OIL in OIL	--	Biological Processes | Dublin Core | Guideline Interchange Format | Resource-Event-Agent Enterprise | Science Ontology | Semantic Translation | Suggested Upper Merged Ontology
CAPABILITIES	**0,1314**			
T11	0,1303	Supported	Supported	Supported
T12	0,1268	Non_Supported	Non_Supported	Non_Supported
T13	0,1324	Non_Supported	Non_Supported	Non_Supported
T14	0,2134	High	High	Very_High
T15	0,1886	Very_High	Very_High	Very_High
T16	0,2085	Low	Very_Low	High
VISUALIZATION	**0,1314**			
T21	0,2651	Supported	Supported	Supported
T22	0,241	Supported	Supported	Supported
T23	0,253	Supported	Supported	Supported
T24	0,241	Supported	Supported	Supported
EDITION	**0,1317**			
T31	0,2748	Non_Supported	Supported	Supported
T32	0,2646	Supported	Supported	Supported
T33	0,2366	Supported	Supported	Supported
T34	0,2239	Non_Supported	Non_Supported	Supported
INTERACTION	**0,119**			
T41	0,189	Supported	Supported	Supported
T42	0,2025	Non_Supported	Non_Supported	Supported
T43	0,2228	Very_Low	Very_Low	High
T44	0,1851	Non_Supported	Non_Supported	Supported
T45	0,2006	Very_Low	Very_Low	High
METHODOLOGICAL ASPECTS	**0,1068**			
T51	0,1811	Non_Supported	Non_Supported	Non_Supported
T52	0,2593	Non_Supported	Non_Supported	Supported
T53	0,2551	Non_Supported	Non_Supported	Supported
T54	0,3045	Supported	Supported	Supported
COOPERATIVE ASPECTS	**0,1154**			
T61	0,1698	Non_Supported	Non_Supported	Non_Supported
T62	0,1456	Non_Supported	Non_Supported	Non_Supported
T63	0,1402	Non_Supported	Non_Supported	Non_Supported
T64	0,1833	Non_Supported	Non_Supported	Non_Supported
T65	0,1779	Non_Supported	Non_Supported	Non_Supported
T66	0,1833	Non_Supported	Non_Supported	Non_Supported
TRANSLATION	**0,1362**			
T71	0,1776	Non_Supported	Supported	Supported
T72	0,1838	Supported	Supported	Supported
T73	0,1501	Non_Supported	Supported	Supported
T74	0,1715	Supported	Supported	Supported
T75	0,1822	Non_Supported	Supported	Non_Supported
T76	0,1348	Non_Supported	Non_Supported	Non_Supported
INTEGRATION	**0,1282**			
T81	0,1609	Very_Low	High	High
T82	0,1666	Very_Low	Low	Low
T83	0,1838	Non_Supported	Non_Supported	Supported
T84	0,1941	Non_Supported	Supported	Supported
T85	0,151	Non_Supported	Non_Supported	Non_Supported
T86	0,1436	Non_Supported	Non_Supported	Non_Supported

Tabla 5.8. Pesos asignados y valores en las instancias candidatas para el caso de prueba 3 sobre entornos de desarrollo de ontologías.

F. MEDIDAS ASIGNADAS POR ONTOMETRIC A LOS CANDIDATOS Y GRÁFICA DE COMPARACIÓN

Instancia	Intervalo obtenido
OilEd	(1.376, 2.460, 5.449, 7.3345)
OntoEdit	(1.907, 3.426, 6.426, 7.902)
Protégé2000	(2.839, 4.906, 7.921, 8.891)

Tabla 5.9. Valores de idoneidad asignados a los entornos candidatos en el caso de prueba 3.

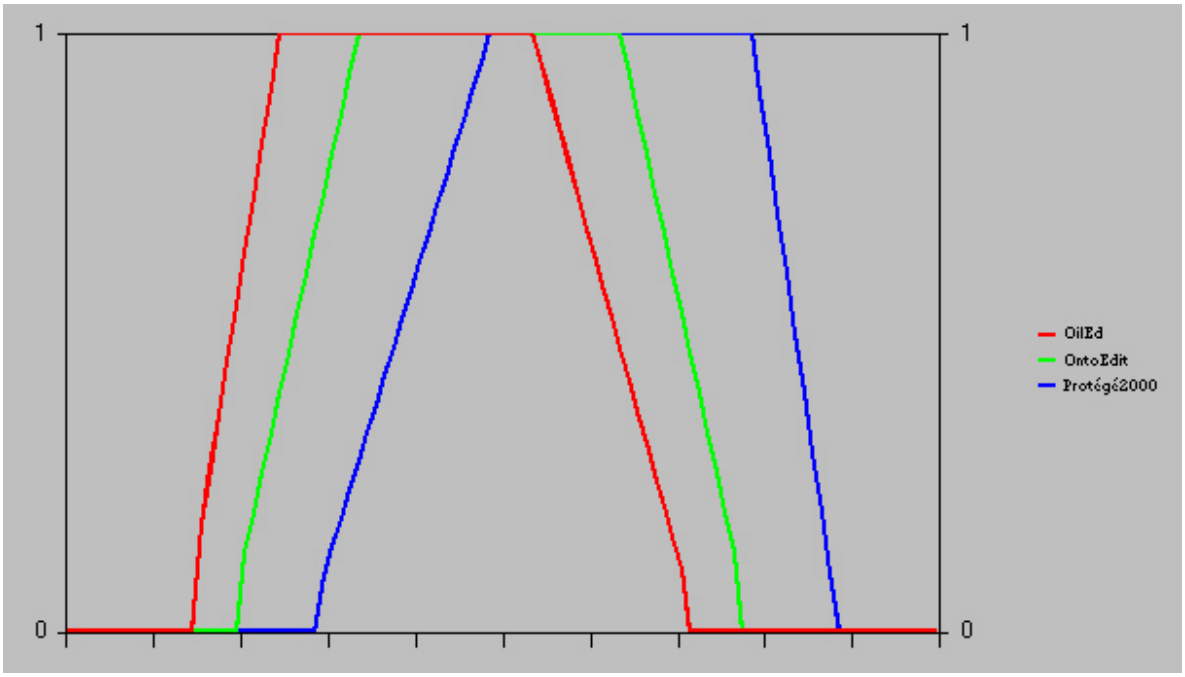


Figura 5.5. Gráfica de comparación mostrada por OT para el caso de estudio 3.

Instancia aconsejada: *Protégé2000*

Entre otros requisitos indicados, se han exigido que los entornos software puedan instalarse localmente (con código T11). Esta ha sido la característica que ha descartado a los entornos *Ontolingua Editor*, *WebOnto* y *WebODE*.

Con los pesos asignados, el mejor valor de idoneidad, como se observa en la tabla 5.9 y en la gráfica de la figura 5.5, lo da *Protégé2000* con un valor cercano a "alto". Si se observa la tabla 5.8, y con los pesos asignados, *Protégé2000* tiene un mayor número de valores positivos que sus competidores *OntoEdit* y *OilEd*, sobre todo en los factores de traducción e integración que, además, están bastante potenciados con los pesos asignados en este caso.

5.2.4. Casos de Prueba para la Dimensión "Contenido de Ontologías"

Como se indicó en la sección 4.5.1, la instanciación de la dimensión "contenido" se ha hecho a partir de ontologías existentes en las direcciones web sobre ontologías y servidores de ontologías: *Ontolingua Server*, *WebODE*, en las página *web* de *SHOE*, del *ITBM-CNR* y de

DAML Ontology Library. Con los *OntoWrappers*, y algunas de forma manual usando *WebODE*, en la RO se completaron las características relacionadas con la dimensión "contenido" que se muestran en la tabla 4.18 de la sección 4.5. Usando los *OntoWrappers*, también se crearon las instancias de relaciones existentes entre los conceptos identificados, que se muestran en la tabla 4.19.

Como ya se explicó en la sección 4.7.1.3.2, el proceso de valoración de idoneidad del "contenido", y también de la dimensión "costes", difiere de los empleados en las otras dimensiones. En estos casos, las valoraciones de las características las realiza el usuario tras examinar la información de las ontologías candidatas proporcionada por la aplicación ROIS. La valoración de idoneidad en cada característica se realiza conforme a una escala lingüística de valoración. La escala lingüística que se ha empleado en este caso de prueba para valorar todas las características, que se muestra en la figura 5.6, consta de 5 valores de idoneidad, aunque podría haberse utilizado otra diferente.

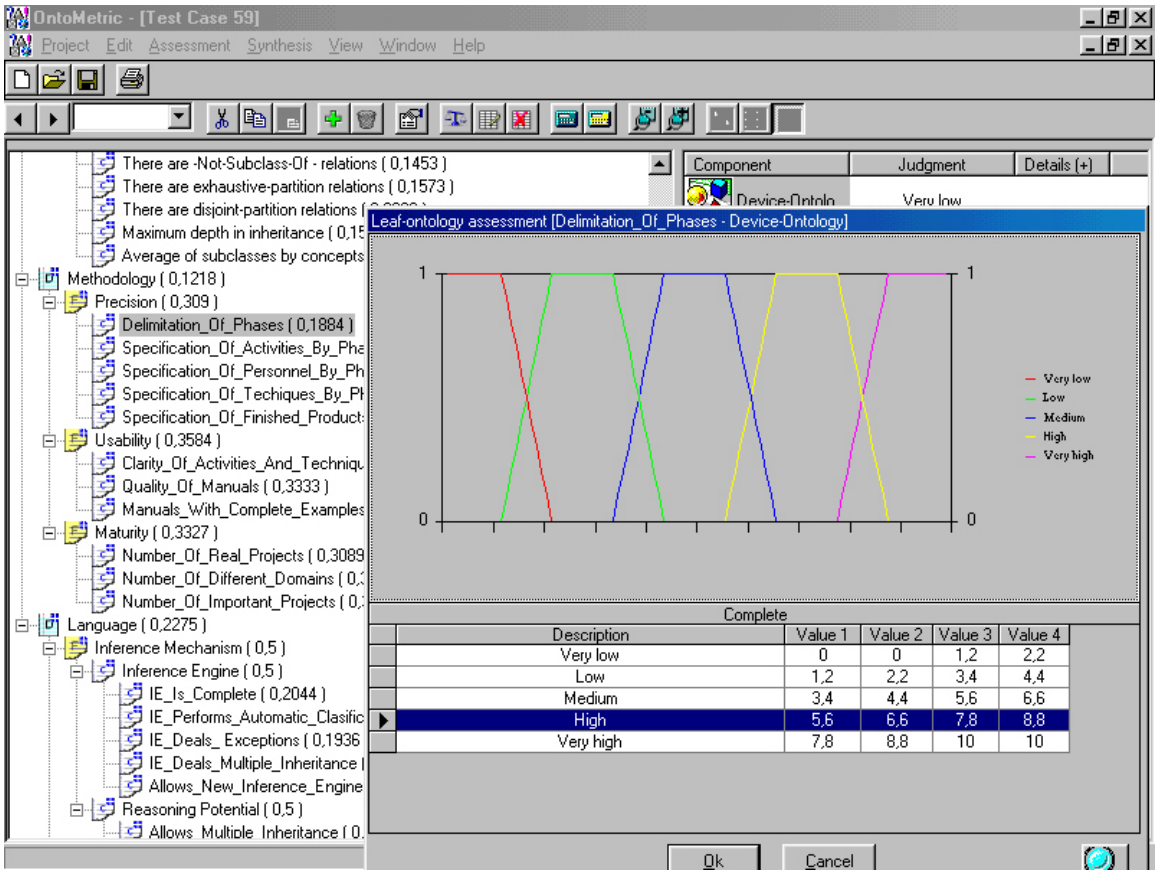


Figura 5.6. Escala lingüística con cinco valores de idoneidad usada en el caso de prueba 4.

En el caso de prueba de este apartado, en el punto 1 se indican los valores en las características descriptivas que deben contener las ontologías candidatas y, también, valores en las características que se exigen sobre los términos de la ontología. Debido a que el número de ontologías es bastante grande, la obtención de las ontologías candidatas esperadas del punto 2

ha consumido una importante cantidad de tiempo. Se debe recordar que el punto 2 sólo se ha realizado en los casos de prueba para comprobar que los valores esperados son los que ROIS obtiene como candidatos. En una aplicación real del método *OntoMetric* no es necesario realizar este paso, ya que la aplicación ROIS, como se verá, tiene un índice muy elevado de aciertos.

En el punto 3 se muestra el contenido de las ontologías encontradas por ROIS, se exponen los valores de las características descriptivas de cada una de las ontologías candidatas, y unos ejemplos sobre cómo ROIS muestra los términos representados de las ontologías seleccionadas. En el punto 4 se exponen los pesos asignado a partir del módulo de OT para calcular los pesos de importancia a partir de las matrices de comparación de pares. En el punto 5, se exponen los valores de idoneidad que se han obtenido en cada uno de ellos, a partir de los valores mostrados en el punto 3. En el punto 6 se representan los valores de los intervalos difusos obtenidos para cada una de las ontologías candidatas examinadas, y una gráfica comparativa sobre los resultados.

CASO DE ESTUDIO 4 (SELECCIÓN DE ONTOLOGÍAS A PARTIR DE LA DIMENSIÓN CONTENIDO):

A. SELECCIÓN DE CARACTERÍSTICAS Y ASIGNACIÓN DE VALORES

En este supuesto se va a buscar una ontología de dominio para realizar el organigrama de una empresa de ventas. Como se muestra en la tabla 5.10, en la primera fila se ha hecho una consulta disyuntiva del término *"organization"* en el nombre de clases y en los nombres de los atributos de las clases existentes. En la segunda fila, de igual forma, en el término *"agent"*. Además, las ontologías deben tener un número de clases entre 5 y 30, deben ser ontologías terminadas, y tienen que ser ontologías de dominio [vHe97]. Es decir, las ontologías que se encuentren deben cumplir todos estos requisitos:

Código	Característica	Valor de búsqueda
CC1, CA1	Name of the class **OR** Name of the attribute	***Organization***
CC1, CA1	Name of the class **OR** Name of the attribute	***Agent***
C019	The number of concepts (classes) represented in the ontology	***C019>5 AND C019<30***
C05	Status of the development of the ontology	***Finished***
C06	Type of ontology represented (van Heist classification)	***Domain***

Tabla 5.10. Valores exigidos al contenido de las ontologías en el caso de prueba 4.

B. CANDIDATOS ESPERADOS

- Agents (Ontolingua)
- Shoe-gen (DAML+OIL)

C. CANDIDATOS ENCONTRADOS POR *ROIS*

- Agents (Ontolingua)
- Shoe-gen (DAML+OIL)

INFORMACIÓN DE LAS ONTOLOGÍAS ENCONTRADAS

En este caso, ROIS muestra al usuario información sobre el contenido de las ontologías candidatas de forma secuencial: primero *Agents* y luego *Shoe-gen*.

ONTOLOGÍA: AGENTS

Código	Característica	Valor
	Ontology: AGENT	
CO1	Name of the ontology	Agent
CO2	A short description of the ontology	Not supplied yet.
CO3	The ontology is physically located at these servers	http://www-ksl.stanford.edu:5915
CO4	Local URL for downloading the original ontology	--
CO5	Status of the development of the ontology	Finished
CO6	Type of ontology represented (van Heist classification)	Domain
CO7	Domain: it describes the piece of reality that the builder of the ontology wants to represent.	--
CO8	Level of generality of the ontology	--
CO9	Purpose: It characterizes the different uses of the ontology.	--
CO10	Formality: It expresses the level of formality of the ontology (Uschold scale).	Very_High
CO11	List of ontologies integrated in the ontology	Frame-Ontology | Slot-Contraint-Sugar
CO12	Kind of formalism paradigms used to formalize the ontology	Frame-Based
CO13	Companies or entities which have developed the ontology	--
CO14	Knowledge sources used in the knowledge acquisition process	--
CO15	Strategy used to identifying the terms	--
CO16	Announced release date of the developed ontology	--
CO17	Date of the last modification of the ontology (changing terms)	--
CO18	List of important projects which use the ontology	--
CO19	The number of concepts (classes) represented in the ontology	6
CO20	The number of instances of concepts represented in the ontology	0
CO21	Average number of attributes in concepts	1
CO22	The number of relations defined in the ontology	1
CO23	The number of functions defined in the ontology	1
CO24	The concepts of the ontology are classified from several perspectives. A class or an instance can have several father classes.	Single
CO25	Number of instances of the relation Subclass-of (or another similar one)	6
CO26	Number of instances of the relation Not-Subclass-of (or another similar one)	0
CO27	Number of exhaustive partition is used in the ontology	0
CO28	Number of disjoint partition is used in the ontology	1
CO29	The number of axioms defined as independent elements (those not linked to concepts).	0
CO30	There are defined axioms to solve queries	False
CO31	There are defined axioms to infer knowledge about attributes of concepts.	False
CO32	There are defined axioms used to verify the consistency of terms in the ontology	False
CO33	The acquisition price of the ontology to be used in the system	0

	CLASS: AGENT (ONTOLOGY:AGENTS)	
CC1	Name of the class	Agent
CC2	Description of the class	An agent is something or someone that can act on its own and produce changes in the world. There is more to agenthood than that, but for this ontology that is all that matters.
CC3	Formal description of the class (literal specification)	(Define-Frame Agent :Own-Slots ((Arity 1) (Documentation "An agent is something or someone that can act on its own and produce changes in the world. There is more to agenthood than that, but for this ontology that is all that matters.") (Instance-Of Class) (Subclass-Of) (Subclass-Partition :Value (Setof Person Organization)) (Subclass-Of Individual-Thing) (Superclass-Of Organization Person)) :Template-Slots ((Name (Cardinality 1))))
R3	ClassIsSubclassOf	Individual-Thing

	ATTRIBUTE: NAME (ONTOLOGY:AGENTS)	
CA1	Name of the attribute	Name
CA2	Description of the attribute	--
CA3	Type of attribute	Instance, Local
CA4	Formal description of the attribute (literal specification)	:Template-Slots ((Name (Cardinality 1))
R5	IsAttributeOfClass	Agent

	CLASS: AGENT-NAME (ONTOLOGY:AGENTS)	
CC1	Name of the class	Agent -Name
CC2	Description of the class	A string that is the name of some agent.
CC3	Formal description of the class (literal specification)	(Define-Frame Agent-Name :Own-Slots ((Arity 1) (Documentation "A string that is the name of some agent.") (Instance-Of Class) (Subclass-Of Individual-Thing)))
R3	ClassIsSubclassOf	Individual-Thing

	CLASS: ORGANIZATION (ONTOLOGY:AGENTS)	
CC1	Name of the class	Organization
CC2	Description of the class	An organization is a corporate or similar institution, distinguished from persons and other agents.
CC3	Formal description of the class (literal specification)	(Define-Class Organization (?X) "An organization is a corporate or similar institution, distinguished from persons and other agents." :Def (Agent ?X))
R3	ClassIsSubclassOf	Agent

• • •

	RELATION: HAS-NAME (ONTOLOGY:AGENTS)	
CR1	Name of the relation	Has-name
CR2	Description of the relation	A mapping from anything to a name for that object. Note that ?x can have multiple names, so this is distinct from Name.
CR3	It is a function	False
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	(Define-Relation Has-Name (?X ?Name) "A mapping from anything to a name for that object. Note that ?x can have multiple names, so this is distinct from Name." :Axiom-Def (=> (= (Name ?X) ?Name) (Has-Name ?X ?Name)))
R9	RelatesClasses	--

	RELATION: NAME (ONTOLOGY:AGENTS)	
CR1	Name of the relation	Name
CR2	Description of the relation	Function from an agent or organisation to the name by which it goes. If an agent has more than one complete name (not parts of the name, such as first and last name), then the NAME is the name used to identify that agent in the shared world. If the shared world is represented in a database, then the name would be a `key field' for the agent. Other names can be related to the agent by some other relations
CR3	It is a function	True
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	(Define-Function Name (?X) "Function from an agent or organisation to the name by which it goes. If an agent has more than one complete name (not parts of the name, such as first and last name), then the NAME is the name used to identify that agent in the shared world. If the shared world is represented in a database, then the name would be a `key field' for the agent. Other names can be related to the agent by some other relations." :Axiom-Def ((=> (Agent ?X) (=> (= (Name ?X) ?Y) (Agent-Name ?Y)))))
R9	RelatesClasses	Agent | Name | Agent-Name

• • •

ONTOLOGÍA: SHOE-GEN

Código	Característica	Valor
	Ontology: SHOE-GEN	
CO1	Name of the ontology	Shoe-gen
CO2	A short description of the ontology	A subset of the Shoe General Ontology
CO3	The ontology is physically located at these servers	http://ubot.lockheedmartin.com/ubot/2000/10/baby-shoe/shoegen-ont.daml
CO4	Local URL for downloading the original ontology	--
CO5	Status of the development of the ontology	Finished
CO6	Type of ontology represented (van Heist classification)	Domain
CO7	Domain: it describes the piece of reality that the builder of the ontology wants to represent.	--
CO8	Level of generality of the ontology	--
CO9	Purpose: It characterizes the different uses of the ontology.	--
CO10	Formality: It expresses the level of formality of the ontology (Uschold scale).	Very_High
CO11	List of ontologies integrated in the ontology	--
CO12	Kind of formalism paradigms used to formalize the ontology	Frame-Based | Description_logic
CO13	Companies or entities which have developed the ontology	--
CO14	Knowledge sources used in the knowledge acquisition process	--
CO15	Strategy used to identifying the terms	--
CO16	Announced release date of the developed ontology	--
CO17	Date of the last modification of the ontology (changing terms)	2000/10/27
CO18	List of important projects which use the ontology	--
CO19	The number of concepts (classes) represented in the ontology	29
CO20	The number of instances of concepts represented in the ontology	0
CO21	Average numb er of attributes in concepts	3
CO22	The number of relations defined in the ontology	17
CO23	The number of functions defined in the ontology	1
CO24	The concepts of the ontology are classified from several perspectives. A class or an instance can have several father classes.	Multiple
CO25	Number of instances of the relation Subclass-of (or another similar one)	29
CO26	Number of instances of the relation Not-Subclass-of (or another similar one)	0
CO27	Number of exhaustive partition is used in the ontology	0
CO28	Number of disjoint partition is used in the ontology	0
CO29	The number of axioms defined as independent elements (those not linked to concepts).	0
CO30	There are defined axioms to solve queries	False
CO31	There are defined axioms to infer knowledge about attributes of concepts.	False
CO32	There are defined axioms used to verify the consistency of terms in the ontology	False
CO33	The acquisition price of the ontology to be used in the system	0

	CLASS: AGENT (ONTOLOGY: SHOE-GEN)	
CC1	Name of the class	Agent
CC2	Description of the class	--
CC3	Formal description of the class (literal specification)	<Class rdf:ID="Agent"> <subClassOf rdf:resource="#Thing" /> </Class>
R3	ClassIsSubclassOf	Thing

	CLASS: ORGANIZATION (ONTOLOGY: SHOE-GEN)	
CC1	Name of the class	Organization
CC2	Description of the class	--
CC3	Formal description of the class (literal specification)	<Class rdf:ID="Organization"> <subClassOf rdf:resource="#SocialGroup" /> </Class>
R3	ClassIsSubclassOf	SocialGroup

	ATTRIBUTE: ORGPHONE (ONTOLOGY: SHOE-GEN)	
CA1	Name of the attribute	OrgPhone
CA2	Description of the attribute	--
CA3	Type of attribute	Instance, Local
CA4	Formal description of the attribute (literal specification)	<Property rdf:ID="orgPhone"> <domain rdf:resource="#Organization" /> </Property>
R5	IsAttributeOfClass	Organization

• • •

	RELATION: AFFILIATEDORGANIZATION (ONTOLOGY: SHOE-GEN)	
CR1	Name of the relation	affiliatedOrganization
CR2	Description of the relation	--
CR3	It is a function	False
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	<Property rdf:ID="affiliatedOrganization"> <domain rdf:resource="#Organization" /> <range rdf:resource="#Organization" /> </Property>
R9	RelatesClasses	Organization | Organization

	RELATION: ALUMNUS (ONTOLOGY: SHOE-GEN)	
CR1	Name of the relation	Alumnus
CR2	Description of the relation	--
CR3	It is a function	False
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	<Property rdf:ID="alumnus"> <domain rdf:resource="#Organization" /> <range rdf:resource="#Person" /> </Property>
R9	RelatesClasses	Organization | Person

. . .

D. y E. PESOS ASIGNADOS Y VALORES DE CARACTERÍSTICAS EN LA DIMENSIÓN "CONTENIDO" DE LAS ONTOLOGÍAS ENCONTRADAS

Código	Peso	Característica	Instancia1(Agents)	Instancia2(Shoe-gen)
	0.3441	***CONCEPTS***		
C11	0.136	The essential concepts for the system are in the ontology	Low	High
C12	0.161	The essential concepts for the system are in the superior levels of the ontology	Low	Very_High
C13	0.152	The concepts are properly described in natural language	High	Very_Low
C14	0.114	The formal specification of each concept coincides with the description of each concept in natural language	Medium	Very_Low
C15	0.226	The attributes of the concepts accurately describe these concepts	Low	High
C16	0.208	The number of concepts represented in the ontology is appropriate for the system	Low	High
	0.2209	*RELATIONS*		
C21	0.205	The essential relations for the system are defined in the ontology	Very_Low	High
C22	0.156	The relations between concepts are those it needed for the system	Very_Low	Medium
C23	0.104	The formal specification of each relation coincides with the description of each relation in natural language	Very_High	Very_Low
C24	0.156	Arity specified in relations is appropriate for the system	High	High
C25	0.156	Formal properties of relations, such as reflexive, irreflexive, symmetrical, asymmetrical, antisymmetrical, transitive and intransitive, are specified in relations appropriately for the system	Very_Low	Very_Low
C26	0.221	The number of relations defined in the ontology is appropriate for the system	Very_Low	High
	0.2928	***TAXONOMY***		
C31	0.1573	The concepts of the ontology are classified from several perspectives appropriately for the system	Medium	High
C32	0.1453	There are "Not-Subclass-Of " relations which are needed for the system	Very_Low	Very_Low
C33	0.1573	There are exhaustive-partition relations which are needed for the system	Very_Low	Very_Low
C34	0.2296	There are disjoint-partition relations which are needed for the system	High	Very_Low
C35	0.1573	The maximum depth in the inheritance of concepts is appropriate for the system	Low	High
C36	0.1531	The average of subclasses by concepts is appropriate for the system	Low	High
	0.1423	***AXIOMS***		
C41	0.1759	There are defined axioms to solve queries	Very_Low	Very_Low
C42	0.3074	There are defined axioms to infer knowledge about attributes of concepts	Very_Low	Very_Low
C43	0.1540	There are defined axioms used to verify the consistency of terms in the ontology.	Very_Low	Very_Low
C44	0.1759	There are defined axioms as independent elements (those not linked to concepts)	Very_Low	Very_Low
C45	0.1868	The number of defined axioms in the ontology is appropriate for the system	Very_Low	Very_Low

Tabla 5.11 Pesos asignados y valores en las ontologías candidatas para el caso de prueba 4 sobre el "contenido" de ontologías.

Los pesos que aparecen en la tabla 5.11 se han obtenido mediante el proceso de cálculo desde las matrices de comparación de pares, implementado en OT.

F. MEDIDAS ASIGNADAS POR ONTOMETRIC A LOS CANDIDATOS Y GRÁFICA DE COMPARACIÓN

Instancia	Intervalo obtenido
Agents	(1.745, 2.351, 3.551, 4.528)
Shoe-gen	(2.865, 3.368, 4.568, 5.512)

Tabla 5.12. Valores de idoneidad asignados al "contenido" de las ontologías candidatas en el caso de prueba 4.

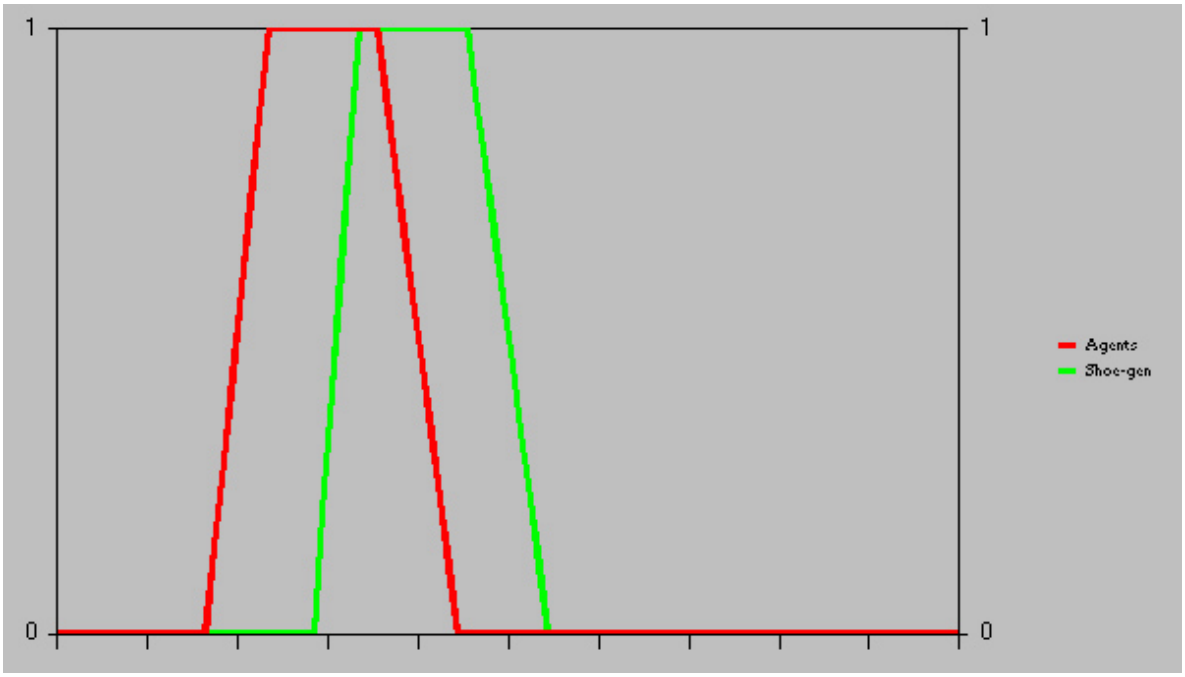


Figura 5.7. Gráfica de comparación mostrada por OT para el caso de estudio 4.

Ontología aconsejada: *Shoe-gen*

Las dos instancias de ontologías que han superado el proceso de selección realizado por ROIS han sido: *Agents* (implementada en *Ontolingua*) y *Shoe-gen* (implementada en *DAML+OIL*). Como se observa en la tabla 5.12 y en la figura 5.7, ambas han obtenido una medida bastante pobre en la dimensión "contenido", lo que indica que sus términos y su organización no coinciden con las pretensiones del supuesto proyecto. La mejor valoración la da *Shoe-gen*, con un valor entre "bajo" y "medio", frente a *Agents* con un valor algo menor que "bajo". Esta diferencia radica fundamentalmente en los conceptos y relaciones de *Agents*, que son muy pocos y, además, no están relacionados con el dominio buscado. El valor de idoneidad calculado mediante *OntoMetric* indica que se deberá realizar un importante esfuerzo de adaptación de la ontología para usarla en el proyecto.

5.3. CASOS DE PRUEBA PARA EVALUAR LAS DIMENSIONES DE FORMA CONJUNTA

En este caso de prueba se usa *OntoMetric* para valorar conjuntamente todas las dimensiones que deben ser examinadas para calcular la idoneidad de las ontologías candidatas. Los pesos asignados en las características para todas las dimensiones se han calculado a partir de las matrices de comparación de pares usando OT.

CASO DE ESTUDIO 5 (SELECCIÓN DE ONTOLOGÍAS A PARTIR DE LAS DIMENSIONES DEL MARCO MULTINIVEL DE CARACTERÍSTICAS):

A. SELECCIÓN DE CARACTERÍSTICAS Y ASIGNACIÓN DE VALORES

El supuesto que se va a realizar en este caso de prueba es valorar la idoneidad de ontologías existentes para integrarlas en un sistema que se va a construir para una empresa editorial de textos científicos. Los elementos de búsqueda que se van a indicar, en varias características de diferentes dimensiones, son los de la tabla 5.13:

Código	Característica	Valor de búsqueda
CC1 , CC2	Name of the class **OR** Description of the class	*Document*
CC1, CA1	Name of the class **OR** Name of the attribute	*Title*
C019	The number of concepts (classes) represented in the ontology	*>5 AND <30*
C05	Status of the development of the ontology	*Finished*
C06	Type of ontology represented (van Heist classification)	*Domain*
L112	The language has metaclasses	*Supported*
L143	The language allows us to define ad-hoc relations	*Supported*
L144	The language allows us to constrain the type of arguments in relations	*Supported*
L151	The language has the relation "Subclass-Of" (or another similar one)	*Supported*
T21	It allows us to visualize all the information that is specified in the ontology	*Supported*
T41	The ontology can be exported from the server and used locally	*Supported*
T54	The tool verifies the changes that have been carried out in the terms of the ontology. They check the consistency of the ontology.	*Supported*
T81	The tool allows easily the integration of ontologies	*High*
Cost1	The acquisition price of the ontology to be used in the system	*0*

Tabla 5.13. Valores exigidos a todas las dimensiones en el caso de prueba 5.

B. CANDIDATOS ESPERADOS

- Bibliographic-Data (Ontolingua)
- Documents (Ontolingua)
- Document (DAML+OIL)

C. ENCONTRADO POR *ROIS*

- Bibliographic-Data (Ontolingua)
- Documents (Ontolingua)
- Document (DAML+OIL)

El ejemplo utilizado en este caso sirve para describir, también, la forma en la que ROIS realiza la búsqueda: en primer lugar, se hace una búsqueda en la dimensión “contenido”. Las ontologías encontradas han sido: *Bibliographic-Data* (*Ontolingua*), *Documents* (*Ontolingua*), y *Document* (*DAML+OIL*). Luego se realiza una búsqueda en la dimensión “metodologías” (en este caso de prueba no se ha exigido ninguna característica a la metodología de desarrollo, por lo cual las

ontologías identificadas en el paso anterior siguen siendo candidatas válidas). De igual forma, se realiza la selección sobre características (L112, L143, L144, L151) exigidas a la dimensión "lenguaje". Posteriormente, se hace la misma operación con las características requeridas (T21, T41, T54, T81) a la dimensión "entornos" que, además, serán los que puedan importar el lenguaje en el que está implementada. Y por último a las características exigidas a la dimensión "costes de uso" (Cost1). El resultado conjunto ha sido el siguiente:

Las ontologías encontradas están implementadas en *Ontolingua* (*Bibliographic-Data* y *Documents*) y *DAML+OIL* (*Document*). Las características exigidas a estos lenguajes (L112, L143, L144, L151) las cumplen *Ontolingua* y *DAML+OIL*, por lo que se mantienen las 3 ontologías candidatas.

Los entornos software que permiten importar los lenguajes en los que están implementadas las tres ontologías (es decir, los entornos que importan *Ontolingua* o *DAML+OIL*) son: Ontolingua Editor (*Ontolingua*), *OntoEdit* (*DAML+OIL*), *OilEd* (*DAML+OIL*) y *Protégé2000* (*Ontolingua*). De entre ellos, los que superan los requisitos indicados (T21, T41, T54, T81) son únicamente *Ontolingua Editor* (*Ontolingua*), *OntoEdit* (*DAML+OIL*), y *Protégé2000* (*Ontolingua*).

Así, tenemos al menos un entorno que puede importar las ontologías *Bibliographic-Data* (*Ontolingua*), *Documents* (*Ontolingua*), y *Document* (*DAML+OIL*). Si por ejemplo, el entorno *OntoEdit* no hubiera superado las exigencias en las características exigidas a los entornos, no se podría trabajar con las ontologías en *DAML+OIL*, y habría que descartar como ontología candidata a la ontología *Document* (*DAML+OIL*).

Por último, todas las ontologías superan la exigencia sobre la característica Cost1 de la dimensión "costes", ya que todas tienen costes de licencia nulo.

D. VALORES EN LAS CARACTERÍSTICAS DE LAS ONTOLOGÍAS ENCONTRADAS EN LAS CINCO DIMENSIONES

En este caso se muestra la información de las ontologías candidatas encontradas, en todas las características de las cinco dimensiones.

CONTENIDO

En este apartado se muestran valores en las características descriptivas de cada ontología candidata, listadas en forma secuencial: primero *Bibliographic-Data*, luego *Documents* y por último *Document.* Debido a su extensión, se exponen únicamente algunos ejemplos del "contenido" de las ontologías que muestra la aplicación ROIS.

INFORMACIÓN DE LAS ONTOLOGÍAS ENCONTRADAS

ONTOLOGÍA: Bibliographic-Data

Código	Característica	Valor
	Ontology: Bibliographic-Data	
CO1	Name of the ontology	Bibliographic-Data
CO2	A short description of the ontology	The bibliographic-data ontology defines the terms used for describing bibliographic references. This ontology defines the basic class for reference objects and the types (classes) for the data objects that appear in references, such as authors and titles …
CO3	The ontology is physically located at these servers	http://www-ksl.stanford.edu:5915
CO4	Local URL for downloading the original ontology	--
CO5	Status of the development of the ontology	Finished
CO6	Type of ontology represented (van Heist classification)	Domain
CO7	Domain: it describes the piece of reality that the builder of the ontology wants to represent.	--
CO8	Level of generality of the ontology	--
CO9	Purpose: It characterizes the different uses of the ontology.	--
CO10	Formality: It expresses the level of formality of the ontology (Uschold scale).	Very_High
CO11	List of ontologies integrated in the ontology	Simple-Time | Agents | Frame-Ontology | Slot-Contraint-Sugar | Documents
CO12	Kind of formalism paradigms used to formalize the ontology	Frame-Based
CO13	Companies or entities which have developed the ontology	Thomas Gruber
CO14	Knowledge sources used in the knowledge acquisition process	--
CO15	Strategy used to identifying the terms	--
CO16	Announced release date of the developed ontology	--
CO17	Date of the last modification of the ontology (changing terms)	--
CO18	List of important projects which use the ontology	--
CO19	The number of concepts (classes) represented in the ontology	33
CO20	The number of instances of concepts represented in the ontology	0
CO21	Average number of attributes in concepts	3
CO22	The number of relations defined in the ontology	21
CO23	The number of functions defined in the ontology	25
CO24	The concepts of the ontology are classified from several perspectives. A class or an instance can have several father classes.	Single
CO25	Number of instances of the relation Subclass-of (or another similar one)	23
CO26	Number of instances of the relation Not-Subclass-of (or another similar one)	0
CO27	Number of exhaustive partition is used in the ontology	1
CO28	Number of disjoint partition is used in the ontology	6
CO29	The number of axioms defined as independent elements (those not linked to concepts).	0
CO30	There are defined axioms to solve queries	False
CO31	There are defined axioms to infer knowledge about attributes of concepts.	False
CO32	There are defined axioms used to verify the consistency of terms in the ontology	False
CO33	The acquisition price of the ontology to be used in the system	0

	CLASS: Biblio-Text (ONTOLOGY: Bibliographic-Data)	
CC1	Name of the class	Biblio-Text
CC2	Description of the class	The most general class of undifferentiated text objects.
CC3	Formal description of the class (literal specification)	(Define-Class Biblio-Text (?X) "The most general class of undifferentiated text objects." :Def (And (Individual-Thing ?X) (Has-Some ?X Has-Textual-Representation)))
R3	ClassIsSubclassOf	Individual-Thing

	CLASS: Cartographic-Map-Reference (ONTOLOGY: Bibliographic-Data)	
CC1	Name of the class	Cartographic-Map-Reference
CC2	Description of the class	A reference to a map created by a cartographer
CC3	Formal description of the class (literal specification)	(Define-Class Cartographic-Map-Reference (?Ref) "A reference to a map created by a cartographer." :Def (And (Misc-Publication-Reference ?Ref) (Inherits-Author-From-Document ?Ref) (Inherits-Title-From-Document ?Ref) (Inherits-Year-From-Document ?Ref)))
R3	ClassIsSubclassOf	Misc-Publication-Reference

• • •

	ATTRIBUTE: Title-of (ONTOLOGY: Bibliographic-Data)	
CA1	Name of the attribute	Cartographic-Map-Reference
CA2	Description of the attribute	
CA3	Type of attribute	Local | Instance
CC3	Formal description of the attribute (literal specification)	(Define-Class Publication-Reference (?Ref) "A reference associated with some kind of published document, where publication and documenthood are interpreted broadly." :Def (And (Reference ?Ref) (Has-One-Of-Type Ref.Document ?Ref Document) (Has-One ?Ref **Title-Of**)) :Axiom-Def (Subclass-Partition Publication-Reference (Setof Book-Reference Book-Section-Reference Article-Reference Proceedings-Paper-Reference Thesis-Reference Technical-Report-Reference Misc-Publication-Reference)))
R5	IsAttributeOfClass	Publication-Reference

• • •

	RELATION: Has-Textual-Representation (ONTOLOGY: Bibliographic-Data)	
CR1	Name of the relation	Has-Textual-Representation
CR2	Description of the relation	The textual representation of a text object. This is a relation rather than a function because the textual representation of a thing could have manifestations in multiple languages or formats
CR3	It is a function	False
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	(Define-Relation Has-Textual-Representation (?X ?String) "The textual representation of a text object. This is a relation rather than a function because the textual representation of a thing could have manifestations in multiple languages or formats." :Def (And (Individual-Thing ?X) (String ?String)))
R9	RelatesClasses	Individual-Thing | String

	RELATION: Ref.Volume (ONTOLOGY: Bibliographic-Data)	
CR1	Name of the relation	Ref.Volume
CR2	Description of the relation	in a reference, the volume number of a journal or magazine in which an article occurs.
CR3	It is a function	True
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	(Define-Function Ref.Volume (?Ref) :-> ?Number "in a reference, the volume number of a journal or magazine in which an article occurs." :Def (And (Or (Book-Reference ?Ref) (Book-Section-Reference ?Ref) (Article-Reference ?Ref)) (Natural ?Number)))
R9	RelatesClasses	Book-Reference | Book-Section-Reference | Article-Reference | Natural

• • •

ONTOLOGÍA: Documents

Código	Característica	Valor
	Ontology: Documents	
CO1	Name of the ontology	Documents
CO2	A short description of the ontology	Not supplied yet
CO3	The ontology is physically located at these servers	http://www-ksl.stanford.edu:5915
CO4	Local URL for downloading the original ontology	--
CO5	Status of the development of the ontology	Finished
CO6	Type of ontology represented (van Heist classification)	Domain
CO7	Domain: it describes the piece of reality that the builder of the ontology wants to represent.	--
CO8	Level of generality of the ontology	High
CO9	Purpose: It characterizes the different uses of the ontology.	--
CO10	Formality: It expresses the level of formality of the ontology (Uschold scale).	Very_High
CO11	List of ontologies integrated in the ontology	Simple-Time | Agents | Frame-Ontology | Slot-Contraint-Sugar
CO12	Kind of formalism paradigms used to formalize the ontology	Frame-Based
CO13	Companies or entities which have developed the ontology	--
CO14	Knowledge sources used in the knowledge acquisition process	--
CO15	Strategy used to identifying the terms	--
CO16	Announced release date of the developed ontology	--
CO17	Date of the last modification of the ontology (changing terms)	--
CO18	List of important projects which use the ontology	--
CO19	The number of concepts (classes) represented in the ontology	20
CO20	The number of instances of concepts represented in the ontology	0
CO21	Average number of attributes in concepts	1
CO22	The number of relations defined in the ontology	4
CO23	The number of functions defined in the ontology	7
CO24	The concepts of the ontology are classified from several perspectives. A class or an instance can have several father classes.	Single
CO25	Number of instances of the relation Subclass-of (or another similar one)	15
CO26	Number of instances of the relation Not-Subclass-of (or another similar one)	0
CO27	Number of exhaustive partition is used in the ontology	1
CO28	Number of disjoint partition is used in the ontology	3
CO29	The number of axioms defined as independent elements (those not linked to concepts).	0
CO30	There are defined axioms to solve queries	False
CO31	There are defined axioms to infer knowledge about attributes of concepts.	False
CO32	There are defined axioms used to verify the consistency of terms in the ontology	False
CO33	The acquisition price of the ontology to be used in the system	0

	CLASS: Document (ONTOLOGY: Documents)	
CC1	Name of the class	Document
CC2	Description of the class	A document is something created by author(s) that may be viewed, listened to, etc., by some audience. A document persists in material form (e.g., a concert or dramatic performance is not a document). Documents typically reside in libraries
CC3	Formal description of the class (literal specification)	(Define-Class Document (?X) "A document is something created by author(s) that may be viewed, listened to, etc., by some audience. A document persists in material form (e.g., a concert or dramatic performance is not a document). Documents typically reside in libraries." :Def (And (Individual-Thing ?X) (Has-One ?X Title-Of)) :Axiom-Def (Subclass-Partition Document (Setof Book Periodical-Publication Proceedings Thesis Technical-Report Miscellaneous-Publication)))
R3	ClassIsSubclassOf	Individual-Thing

	CLASS: Magazine (ONTOLOGY: Documents)	
CC1	Name of the class	Magazine
CC2	Description of the class	A magazine is a periodical publication that is considered to be of more general interest than a journal
CC3	Formal description of the class (literal specification)	(Define-Class Magazine (?X) "A magazine is a periodical publication that is considered to be of more general interest than a journal." :Def (Periodical-Publication ?X))
R3	ClassIsSubclassOf	Periodical-Publication

...

	RELATION: Has-Series-Editor (ONTOLOGY: Documents)	
CR1	Name of the relation	Has-Series-Editor
CR2	Description of the relation	Series editors of a document.
CR3	It is a function	False
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	(Define-Relation Has-Series-Editor (?Doc ?Editor) "Series editors of a document." :Def (And (Document ?Doc) (Person ?Editor)))
R9	RelatesClasses	Document | Person

	RELATION: Has-Translator (ONTOLOGY: Documents)	
CR1	Name of the relation	Has-Translator
CR2	Description of the relation	Named primary translators of a document
CR3	It is a function	False
CR4	Relation arity	2
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	(Define-Relation Has-Translator (?Doc ?Translator) "Named primary translators of a document." :Def (And (Document ?Doc) (Agent ?Translator)))
R9	RelatesClasses	Document | Translator

...

ONTOLOGÍA: Document

Código	Característica	Valor
	Ontology: Document	
CO1	Name of the ontology	Document
CO2	A short description of the ontology	Ontology about documents.
CO3	The ontology is physically located at these servers	http://www.daml.org
CO4	Local URL for downloading the original ontology	http://www.isi.edu/webscripter/document.o.daml
CO5	Status of the development of the ontology	Finished
CO6	Type of ontology represented (van Heist classification)	Domain
CO7	Domain: it describes the piece of reality that the builder of the ontology wants to represent.	--
CO8	Level of generality of the ontology	--
CO9	Purpose: It characterizes the different uses of the ontology.	--
CO10	Formality: It expresses the level of formality of the ontology (Uschold scale).	Very_High
CO11	List of ontologies integrated in the ontology	Persons | Projects
CO12	Kind of formalism paradigms used to formalize the ontology	Frame-Based | Description_logic
CO13	Companies or entities which have developed the ontology	Pedro Szekely (USC/ISI)
CO14	Knowledge sources used in the knowledge acquisition process	--
CO15	Strategy used to identifying the terms	--
CO16	Announced release date of the developed ontology	--
CO17	Date of the last modification of the ontology (changing terms)	--
CO18	List of important projects which use the ontology	--
CO19	The number of concepts (classes) represented in the ontology	4
CO20	The number of instances of concepts represented in the ontology	0
CO21	Average number of attributes in concepts	1
CO22	The number of relations defined in the ontology	4
CO23	The number of functions defined in the ontology	0
CO24	The concepts of the ontology are classified from several perspectives. A class or an instance can have several father classes.	Single
CO25	Number of instances of the relation Subclass-of (or another similar one)	3
CO26	Number of instances of the relation Not-Subclass-of (or another similar one)	0
CO27	Number of exhaustive partition is used in the ontology	0
CO28	Number of disjoint partition is used in the ontology	0
CO29	The number of axioms defined as independent elements (those not linked to concepts).	0
CO30	There are defined axioms to solve queries	False
CO31	There are defined axioms to infer knowledge about attributes of concepts.	False
CO32	There are defined axioms used to verify the consistency of terms in the ontology	False
CO33	The acquisition price of the ontology to be used in the system	0

	CLASS: Document (ONTOLOGY: Document)	
CC1	Name of the class	Document
CC2	Description of the class	Document. Metadata about a document
CC3	Formal description of the class (literal specification)	<Class rdf:ID="Document"> <label>Document</label> <comment>Metadata about a document.</comment> <subClassOf rdf:resource="#DocumentBase"/> </Class>
R3	ClassIsSubclassOf	DocumentBase

	CLASS: DocumentAlternative (ONTOLOGY: Document)	
CC1	Name of the class	DocumentAlternative
CC2	Description of the class	Alternative format of a document revision. Same contents, just in a different format (e.g., HTML and PDF)
CC3	Formal description of the class (literal specification)	<Class rdf:ID="DocumentAlternative" label="DocumentAlternative" <comment>Alternative format of a document revision. Same contents, just in a different format (e.g., HTML and PDF)</comment> <subClassOf rdf:resource="#DocumentBase"/> </Class>
R3	ClassIsSubclassOf	DocumentBase

• • •

	RELATION: Revision (ONTOLOGY: Document)	
CR1	Name of the relation	Revision
CR2	Description of the relation	A revision of a document.
CR3	It is a function	False
CR4	Relation arity	2
CR5	Formal properties specified	inverse
CR6	Formal description of the relation (literal specification)	<Property rdf:ID="revision"> <domain rdf:resource="#Document"/> <range rdf:resource="#DocumentRevision"/> <comment>A revision of a document.</comment> </Property>
R9	RelatesClasses	Document | DocumentRevision

	RELATION: Format (ONTOLOGY: Document)	
CR1	Name of the relation	Format
CR2	Description of the relation	--
CR3	It is a function	False
CR4	Relation arity	1
CR5	Formal properties specified	--
CR6	Formal description of the relation (literal specification)	<Property rdf:ID="format"> <domain rdf:resource="#DocumentRevision"/> <cardinality>1</cardinality> </Property>
R9	RelatesClasses	DocumentRevision

• • •

D.1. y E.1. PESOS ASIGNADOS Y VALORES DE CARACTERÍSTICAS EN LA DIMENSIÓN "CONTENIDO" DE LAS ONTOLOGÍAS ENCONTRADAS

En la tabla 5.14 aparecen los valores indicados en cada una de las características, y en la tabla 5.15 las medidas obtenidas para esta dimensión.

	CONTENIDO				
Código	**Peso**	**Característica**	**Instancia1 (Bibliographi -Data)**	**Instancia2 (Documents)**	**Instancia3 (Document)**
	0.1723	***CONCEPTS***			
C11	0.1540	The essential concepts for the system are in the ontology	Very_High	Medium	Low
C12	0.1692	The essential concepts for the system are in the superior levels of the ontology	Very_High	Very_High	Low
C13	0.1540	The concepts are properly described in natural language	High	Medium	Low
C14	0.1366	The formal specification of each concept coincides with the description of each concept in natural language	Medium	High	Medium
C15	0.2322	The attributes of the concepts accurately describe these concepts	Low	High	High
C16	0.1540	The number of concepts represented in the ontology is appropriate for the system	Very_High	Medium	Very_Low
	0.3570	RELATIONS			
C21	0.1349	The essential relations for the system are defined in the ontology	High	Medium	Very_Low
C22	0.1513	The relations between concepts are those it needed for the system	Medium	Low	Very_Low
C23	0.1319	The formal specification of each relation coincides with the description of each relation in natural language	High	High	Very_Low
C24	0.1513	Arity specified in relations is appropriate for the system	High	High	High
C25	0.1513	Formal properties of relations, such as reflexive, irreflexive, symmetrical, asymmetrical, antisymmetrical, transitive and intransitive, are specified in relations appropriately for the system	Very_Low	Very_Low	Low
C26	0.2493	The number of relations defined in the ontology is appropriate for the system	High	Medium	Very_Low
	0.2353	***TAXONOMY***			
C31	0.1573	The concepts of the ontology are classified from several perspectives appropriately for the system	Medium	Medium	Medium
C32	0.1453	There are "Not-Subclass-Of " relations which are needed for the system	Very_Low	Very_Low	Very_Low
C33	0.1573	There are exhaustive-partition relations which are needed for the system	Medium	Low	Very_Low
C34	0.2296	There are disjoint-partition relations which are needed for the system	High	Low	Very_Low
C35	0.1573	The maximum depth in the inheritance of concepts is appropriate for the system	High	Medium	Very_Low
C36	0.1531	The average of subclasses by concepts is appropriate for the system	High	High	Low
	0.2353	***AXIOMS***			
C41	0.1759	There are defined axioms to solve queries	Very_Low	Very_Low	Very_Low
C42	0.3074	There are defined axioms to infer knowledge about attributes of concepts	Very_Low	Very_Low	Very_Low
C43	0.1540	There are defined axioms used to verify the consistency of terms in the ontology.	Very_Low	Very_Low	Very_Low
C44	0.1759	There are defined axioms as independent elements (those not linked to concepts)	Very_Low	Very_Low	Very_Low
C45	0.1868	The number of defined axioms in the ontology is appropriate for the system	Very_Low	Very_Low	Very_Low

Tabla 5.14. Pesos asignados y valores en las instancias candidatas sobre el "contenido" para el caso de prueba 5.

F.1. MEDIDAS ASIGNADAS POR *ONTOMETRIC* A LA DIMENSIÓN "CONTENIDO" Y GRÁFICA DE COMPARACIÓN

Instancia	Intervalo obtenido
Bibliographic-Data	(3.510, 4.187, 5.387, 6.304)
Documents	(2.573, 3.250, 4.500, 5.420)
Document	(0.939, 1.265, 2.465, 3.465)

Tabla 5.15. Valores de idoneidad asignados al contenido de las ontologías en el caso de prueba 5.

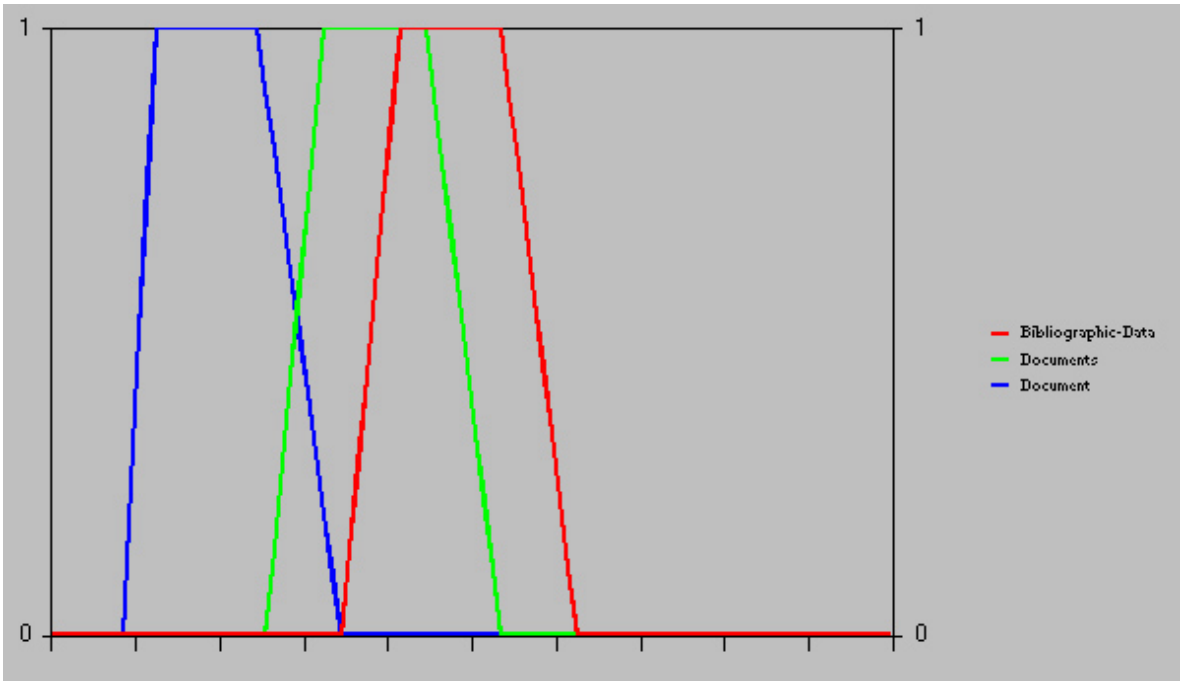


Figura 5.8. Gráfica de comparación para la dimensión "contenido" mostrada por OT para el caso de estudio 5.

En la dimensión de contenido, en la figura 5.8 se observa que la ontología que mejores valores obtiene es *Bibliographic_Data*, debido a que tiene un mayor número de clases y relaciones coincidentes con las necesidades en el proyecto. La peor ontología, *Document*, tiene un escaso número de clases y relaciones, y muestra una medida muy pobre en esta dimensión.

Con estos valores, y si se decide utilizar en el proyecto la ontología *Bibliographic_Data* se deberá realizar un importante proceso de adaptación de los términos, ya que la medida de idoneidad del contenido de esta ontología tiene un valor cercano a "medio".

LENGUAJE

Se exponen en la tabla 5.16 los valores en las características de los lenguajes en los que están implementadas las ontologías candidatas: *Ontolingua*, en el que están implementadas las ontologías *Bibliographic_Data* y *Documents*, y *DAML+OIL*, en el que está implementada la ontología *Document.*

D.2. y E.2. PESOS ASIGNADOS Y VALORES DE CARACTERÍSTICAS EN LA DIMENSIÓN "LENGUAJE"

Código	Característica	Peso	(Bibliographic-Data) (Documents)	(Document)
L1	**DOMAIN KNOWLEDGE**	0 . 5	Ontolingua	DAML+OIL
L11	CONCEPTS/ INSTANCES/ FACTS/ CLAIMS	0.1699		
L111	It allows us to create instances of class	0.2548	Supported	Supported
L112	The language has metaclasses	0.1599	Supported	Supported
L113	Classes can be created without using metaclasses	0.3128	Supported	Supported
L114	The language allows facts	0.1831	Supported	Non_supported
L115	The language allows claims	0.0894	Non_supported	Non_supported
L12	***ATTRIBUTES***	0.1583		
L121	Class Attributes (own slots)	0.18	Supported	Supported
L122	Instance Attributes (template slots)	0.222	Supported	Supported
L123	Local Attributes	0.1501	Supported	Supported
L124	Global Attributes	0.1294	Supported	Supported
L125	Polymorph Attributes	0.1526	Supported	Supported
L126	Language allows us to define exceptions in attributes (redefine attributes)	0.1656	Non_supported	Non_supported
L13	***FACETS***	0.1202		
L131	The facet "Default value" (or another similar one) exists	0.1852	Supported	Non_supported
L132	The facet "Type of value" (or another similar one) exists	0.229	Supported	Supported
L133	The facet "Cardinality" (or similar) exists	0.2279	Supported	Supported
L134	The language allows us to define procedural knowledge	0.1743	Non_supported	Non_supported
L135	The language allows us to define new facets	0.1831	Supported	Non_supported
L14	***RELATIONS***	0.1466		
L141	The language has relations type "function" (those that return an argument)	0.1106	Supported	Non_supported
L142	The language allows us to use relations with arity superior to binary	0.131	Supported	Non_supported
L143	The language allows us to define ad-hoc relations	0.2092	Supported	Supported
L144	The language allows us to constrain the type of arguments in relations	0.1322	Supported	Supported
L145	The language allows us to constrain the value of the arguments in relations	0.1661	Supported	Non_supported
L146	The language allows operational definitions in relations	0.1435	Non_supported	Non_supported
L147	The language allows us to declare mathematical properties in relations	0.1075	Supported	Supported
L15	***TAXONOMY***	0.1497		
L151	The language has the relation "Subclass-Of" (or another similar one)	0.1978	Supported	Supported
L152	The language has the relation "Not-Subclass-Of" (or another similar one)	0.1108	Supported	Non_supported
L153	The language allows us to define exhaustive partitions	0.1585	Supported	Supported
L154	The language allows us to define disjoint partitions	0.1545	Supported	Supported
L155	A class can have several father classes	0.194	Supported	Supported
L156	An instance can have several father classes	0.1845	Non_supported	Supported
L16	***AXIOMS***	0.1311		
L161	The language allows us to declare axioms embedded in terms	0.3016	Supported	Non_supported
L162	The language allows us to declare independent (named) axioms	0.1697	Supported	Non_supported
L163	The language allows us to represent axioms with FOL	0.2668	Supported	Non_supported
L164	The language allows us to represent axioms with second order logic	0.2619	Supported	Non_supported
L17	***PRODUCTION RULES***	0.1243		
L171	The language allows us to use disjuntives in production rules	0.1643	Non_supported	Non_supported
L172	The language allows us to use conjuntives in production rules	0.1113	Non_supported	Non_supported
L173	The language allows us to define the chaining mechanism attached to each rule	0.1621	Non_supported	Non_supported
L174	The language allows us to define the application priority attached to each rule	0.1885	Non_supported	Non_supported
L175	The language allows us to use procedures in the consequent of the rule	0.1857	Non_supported	Non_supported
L176	The language allows us to define certain values attached to each rule	0.1881	Non_supported	Non_supported
L2	**INFERENCE MECHANISM**	0 . 5	Ontolingua	DAML+OIL
L21	***REASONING POTENTIAL***	0.5		
L211	The language allows multiple inheritance	0.1744	Supported	Supported
L212	The language allows monotonous reasoning	0.1426	Supported	Non_supported
L213	The language allows non-monotonous reasoning	0.1213	Non_supported	Non_supported
L214	The redefinition of attributes can be used to make exceptions in inheritance	0.1606	Non_supported	Non_supported
L215	The system can use axioms to maintain its consistency. This is done by checking restrictions.	0.1684	Supported	Non_supported
L216	The system can execute the procedures that are defined in the language	0.1508	Supported	Non_supported
L217	There is an inference mechanism (forward, backward or mixture) linked to the production rules.	0.082	Non_supported	Non_supported
L22	***INFERENCE ENGINE***	0.5		
L221	The inference engine is sound and complete	0.2044	Non_supported	Non_supported
L222	The inference engine performs automatic classifications	0.1625	Non_supported	Non_supported
L223	The inference engine deals with exceptions	0.1936	Non_supported	Non_supported
L224	The inference engine deals with multiple inheritance	0.2424	Non_supported	Non_supported
L225	You can build a new inference engine	0.1972	Supported	Supported

Tabla 5.16. Pesos asignados y valores en las instancias candidatas sobre el "lenguaje" para el caso de prueba 5.

F.2. MEDIDAS ASIGNADAS POR *ONTOMETRIC* A LA DIMENSIÓN "LENGUAJE" Y GRÁFICA DE COMPARACIÓN

La valoración de idoneidad de los lenguajes candidatos se muestra en la tabla 5.17:

Instancia	Intervalo obtenido
Ontolingua (Bibliographic-Data, Documents)	(1.803, 3.607, 6.962, 8.459)
DAML+OIL (Document)	(1.615, 2.323, 5.658, 7.797)

Tabla 5.17. Valores de idoneidad asignados a los lenguajes en el caso de prueba 5.

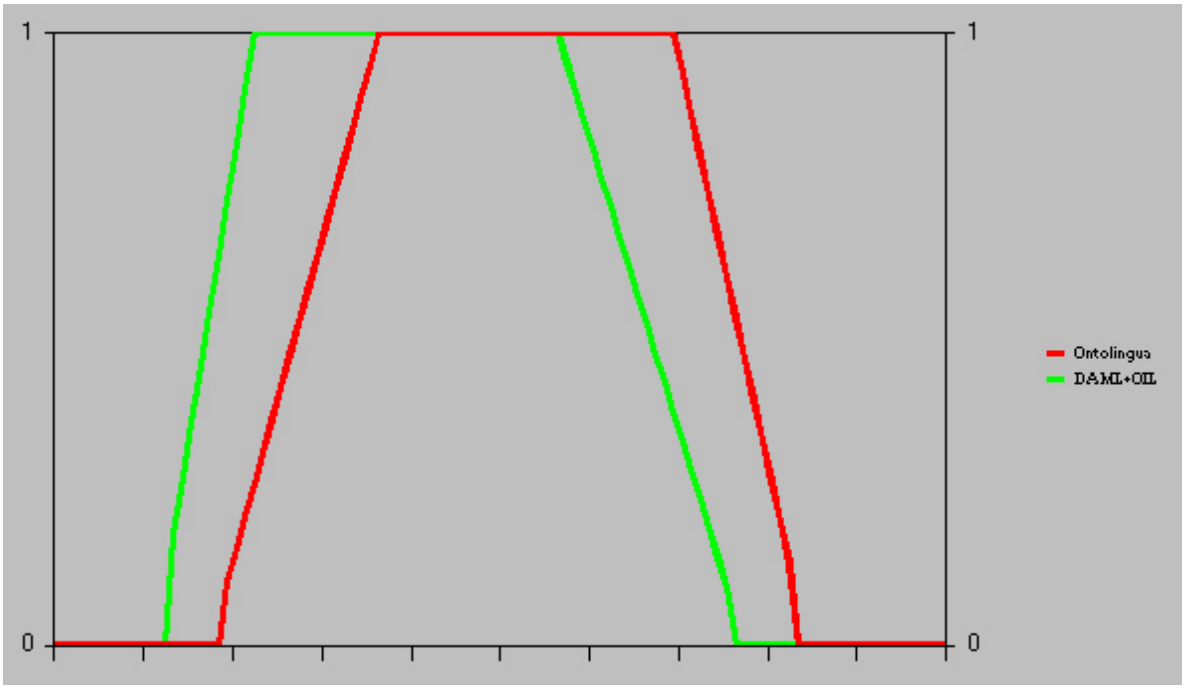


Figura 5.9. Gráfica de comparación para la dimensión "lenguajes" mostrada por OT para el caso de estudio 5.

En este caso, como se observa en la figura 5.9, el lenguaje que mejor valor obtiene es *Ontolingua*, en el que están implementadas las ontologías *Bibliographic_Data* y *Documents*, con respecto a *DAML+OIL* en el que está implementada la ontología *Document*. Si se observa la tabla 5.16, *Ontolingua* posee en la mayoría de las características mejores valores que *DAML+OIL*. Es bastante significativo el conjunto de características relacionadas con la capacidad de representar axiomas.

ENTORNO

En este caso, el entorno *OntoEdit* permite importar la ontología *Document* desde *DAML+OIL* y supera los requisitos indicados (T21, T41, T54, T81) en la fase de selección.

Los entornos *Ontolingua Editor* y *Protégé2000* permiten importar las ontologías *Bibliographic-Data* y *Documents* implementadas en *Ontolingua* y superan los requisitos indicados (T21, T41,

T54, T81) en la fase de selección.

Los pesos que aparecen en la tabla 5.18 se han obtenido mediante el proceso de cálculo desde las matrices de comparación de pares, implementado en OT.

D.3. y E.3. PESOS ASIGNADOS Y VALORES DE CARACTERÍSTICAS EN LA DIMENSIÓN "ENTORNO"

Código	Peso	OntoEdit	Ontolingua editor	Protégé2000
CAPABILITIES	**0,1314**			
T11	0,1303	Supported	Non_Supported	Supported
T12	0,1268	Non_Supported	Non_Supported	Non_Supported
T13	0,1324	Non_Supported	Supported	Non_Supported
T14	0,2134	High	Low	Very_High
T15	0,1886	Very_High	Medium	Very_High
T16	0,2085	Very_Low	Very_High	High
VISUALIZATION	**0,1314**			
T21	0,2651	Supported	Supported	Supported
T22	0,241	Supported	Supported	Supported
T23	0,253	Supported	Supported	Supported
T24	0,241	Supported	Supported	Supported
EDITION	**0,1317**			
T31	0,2748	Supported	Supported	Supported
T32	0,2646	Supported	Supported	Supported
T33	0,2366	Supported	Supported	Supported
T34	0,2239	Non_Supported	Non_Supported	Supported
INTERACTION	**0,119**			
T41	0,189	Supported	Supported	Supported
T42	0,2025	Non_Supported	Non_Supported	Supported
T43	0,2228	Very_Low	Very_Low	High
T44	0,1851	Non_Supported	Non_Supported	Supported
T45	0,2006	Very_Low	Very_Low	High
METHODOLOGICAL ASPECTS	**0,1068**			
T51	0,1811	Non_Supported	Non_Supported	Non_Supported
T52	0,2593	Non_Supported	Supported	Supported
T53	0,2551	Non_Supported	Non_Supported	Supported
T54	0,3045	Supported	Supported	Supported
COOPERATIVE ASPECTS	**0,1154**			
T61	0,1698	Non_Supported	Supported	Non_Supported
T62	0,1456	Non_Supported	Supported	Non_Supported
T63	0,1402	Non_Supported	Supported	Non_Supported
T64	0,1833	Non_Supported	Supported	Non_Supported
T65	0,1779	Non_Supported	Supported	Non_Supported
T66	0,1833	Non_Supported	Non_Supported	Non_Supported
TRANSLATION	**0,1362**			
T71	0,1776	Supported	Supported	Supported
T72	0,1838	Supported	Non_Supported	Supported
T73	0,1501	Supported	Non_Supported	Supported
T74	0,1715	Supported	Non_Supported	Supported
T75	0,1822	Supported	Supported	Non_Supported
T76	0,1348	Non_Supported	Non_Supported	Non_Supported
INTEGRATION	**0,1282**			
T81	0,1609	High	High	High
T82	0,1666	Low	Low	Low
T83	0,1838	Non_Supported	Supported	Supported
T84	0,1941	Supported	Supported	Supported
T85	0,151	Non_Supported	Non_Supported	Non_Supported
T86	0,1436	Non_Supported	Non_Supported	Non_Supported

Tabla 5.18. Pesos asignados y valores en las instancias candidatas sobre el "entorno" para el caso de prueba 5.

F.3. MEDIDAS ASIGNADAS POR *ONTOMETRIC* A LA DIMENSIÓN "ENTORNO" Y GRÁFICA DE COMPARACIÓN

Los valores obtenidos para la dimensión "entorno de desarrollo" se muestran en la tabla 5.19.

Instancia	Intervalo obtenido
OntoEdit (*Document*)	(1.907, 3.426, 6.426, 7.902)
Ontolingua Editor (*Bibliographic_Data* y *Documents*)	(2.146, 4.009, 7.020, 8.177)
Protégé2000 (Bibliographic_Data y Documents)	(2.839, 4.906, 7.921, 8.891)

Tabla 5.19. Valores de idoneidad asignados a los entornos en el caso de prueba 5.

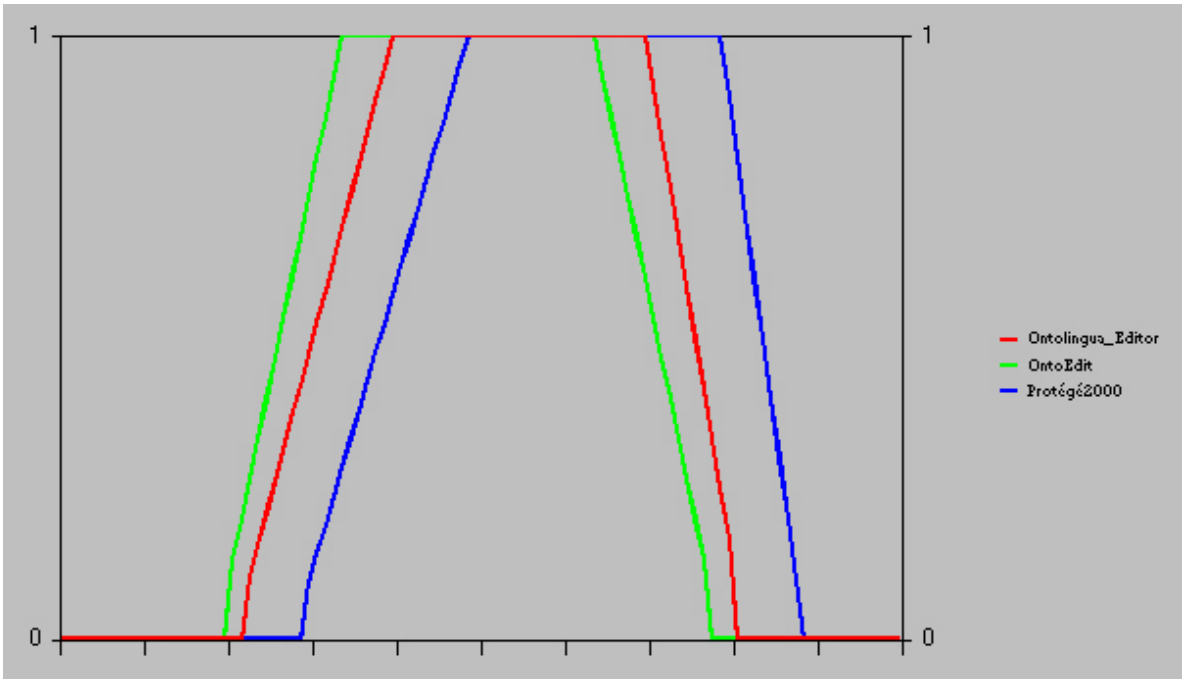


Figura 5.10. Gráfica de comparación para la dimensión "entornos" mostrada por OT para el caso de estudio 5.

Como se puede observar en la figura 5.10, los mejores valores en esta dimensión se obtienen para el entorno *Protégé2000* que es capaz de importar las ontologías *Bibliographic_Data* y *Documents*. Los valores de este entorno se asignarán a estas dos ontologías en el cálculo de valoración global. El entorno *OntoEdit*, asociado con la ontología *Document*, tiene una valoración, con los pesos asignados, bastante peor.

A pesar de que el *Ontolingua Editor* tiene mejores valores que los demás candidatos en el factor "aspectos cooperativos", los tiene muy bajos en el factor "traducción". Como se observa en la tabla 5.18, el entorno *Protégé2000* tiene, entre otras características, unos valores superiores a sus competidores en las características del factor "prestaciones".

METODOLOGÍA

Ninguna de las ontologías candidatas tiene indicada una metodología de desarrollo. En este caso se asignan a todas las características hoja de esta dimensión el menor valor posible, según las escalas asignadas a cada característica respectivamente. Así, el cálculo obtenido con estos valores han producido los intervalos de la tabla 5.20.

F.4. MEDIDAS ASIGNADAS POR *ONTOMETRIC* A LA DIMENSIÓN "METODOLOGÍA"

Instancia	Intervalo obtenido
Ninguna (Bibliographic-Data)	(0, 0, 1.450, 2.7258)
Ninguna (Documents)	(0, 0, 1.450, 2.7258)
Ninguna (Document)	(0, 0, 1.450, 2.7258)

Tabla 5.20. Valores de idoneidad asignados a las metodologías en el caso de prueba 5.

COSTES

Como se observa en la tabla 5.23, todas las ontologías y los entornos software que pueden importarlas tienen similares precios y, por tanto, el valor de idoneidad que se ha obtenido es similar para las tres ontologías. En este caso, se ha decidido asignar el mejor valor a todos los criterios de costes (tabla 5.22) a partir de la información que aparece en la tabla 5.21.

INFORMACIÓN SOBRE LOS COSTES

Código	Característica	Peso	Document (OntoEdit)	Bibliographic-Data (Ontolingua Editor) (Protégé2000)	Documents (Ontolingua Editor) (Protégé2000)
Cost1 (DESDE CO33)	The acquisition price of the ontology to be used in the system	0,25	0	0	0
Cost2 (DESDE T05)	The costs to acquire the hardware and software resources in order to use the ontology	0.25	1000	1200/1000	1200/1000
Cost3 (DESDE T08)	The acquisition price of the access interfaces of the ontology	0.25	0	0	0
Cost4 (DESDE T07)	The acquisition price of the edition and visualization tools of the ontology	0.25	0	0	0

Tabla 5.21. Valores en las características correspondientes que se consideran para obtener la valoración de la dimensión "costes" en el caso de prueba 5.

D.5. y E.5. PESOS ASIGNADOS Y VALORES DE CARACTERÍSTICAS EN LA DIMENSIÓN "COSTES"

Código	Característica	Peso	Document (OntoEdit)	Bibliographic-Data (Ontolingua Editor) (Protégé2000)	Documents (Ontolingua Editor) (Protégé2000)
Cost1	The acquisition price of the ontology to be used in the system	0,25	Very_High	Very_High	Very_High
Cost2	The costs to acquire the hardware and software resources in order to use the ontology	0.25	Very_High	Very_High	Very_High
Cost3	The acquisition price of the access interfaces of the ontology	0.25	Very_High	Very_High	Very_High
Cost4	The acquisition price of the edition and visualization tools of the ontology	0.25	Very_High	Very_High	Very_High

Tabla 5.22. Pesos asignados y valores en las instancias candidatas sobre "coste de uso" en el caso de prueba 5.

F.5. MEDIDAS ASIGNADAS POR *ONTOMETRIC* A LA DIMENSIÓN "COSTES"

Instancia	Intervalo obtenido
Bibliographic-Data (Ontolingua)	(7.8, 8.8, 10, 10)
Documents (Ontolingua)	(7.8, 8.8, 10, 10)
Document (DAML+OIL)	(7.8, 8.8, 10, 10)

Tabla 5.23. Valores de idoneidad asignados a los costes de uso en el caso de prueba 5.

VALORACIÓN MULTIDIMENSIONAL

Por último, con los valores obtenidos en cada dimensión, y representados en los correspondientes intervalos difusos, se calcula el valor de idoneidad de cada una de las ontologías teniendo en cuenta todas las dimensiones en conjunto. Para obtener esta medida, cada uno de los valores de cada dimensión será ponderado por el peso asignado a la dimensión correspondiente. Se han obtenido los siguientes resultados, mostrados en la tabla 5.24 y en la gráfica de la figura 5.11:

Dimensión	Peso	Bibliographic-Data (Ontolingua)	Documents (Ontolingua)	Document (DAML+OIL)
Contenido	0.2304	(3.510, 4.187, 5.387, 6.304)	(2.573, 3.250, 4.500, 5.420)	(0.939, 1.265, 2.465, 3.465)
Lenguaje	0.2275	(1.803, 3.607, 6.962, 8.459)	(1.803, 3.607, 6.962, 8.459)	(1.615, 2.323, 5.658, 7.797)
Metodología	0.1218	(0, 0, 1.450, 2.7258)	(0, 0, 1.450, 2.7258)	(0, 0, 1.450, 2.7258)
Entorno	0.2696	(2.839, 4.906, 7.921, 8.891)	(2.839, 4.906, 7.921, 8.891)	(1.907, 3.426, 6.426, 7.902)
Costes	0.1507	(7.8, 8.8, 10, 10)	(7.8, 8.8, 10, 10)	(7.8, 8.8, 10, 10)
VALORACIÓN DE IDONEIDAD		**(3.295, 4.715, 6.939, 7.763)**	**(3.079, 4.499, 6.723, 7.560)**	**(2.203, 3.139, 5.352, 6.595)**

Tabla 5.24. Valores de idoneidad asignados a cada ontología candidata, tomando en cuenta todas las dimensiones en el caso de prueba 5.

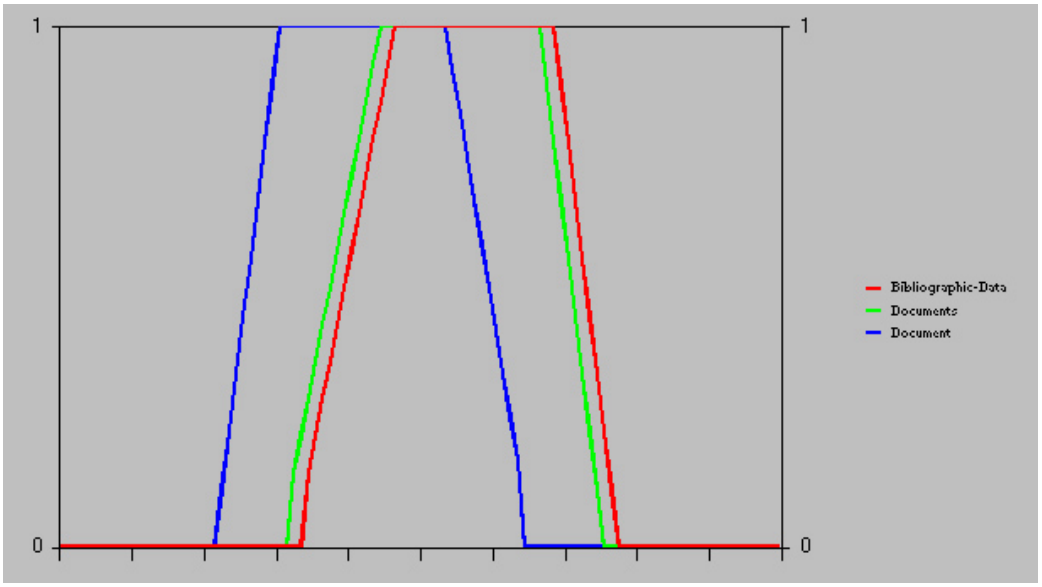


Figura 5.11. Gráfica de comparación multidimensional mostrada por OT para las ontologías candidatas en el caso de estudio 5.

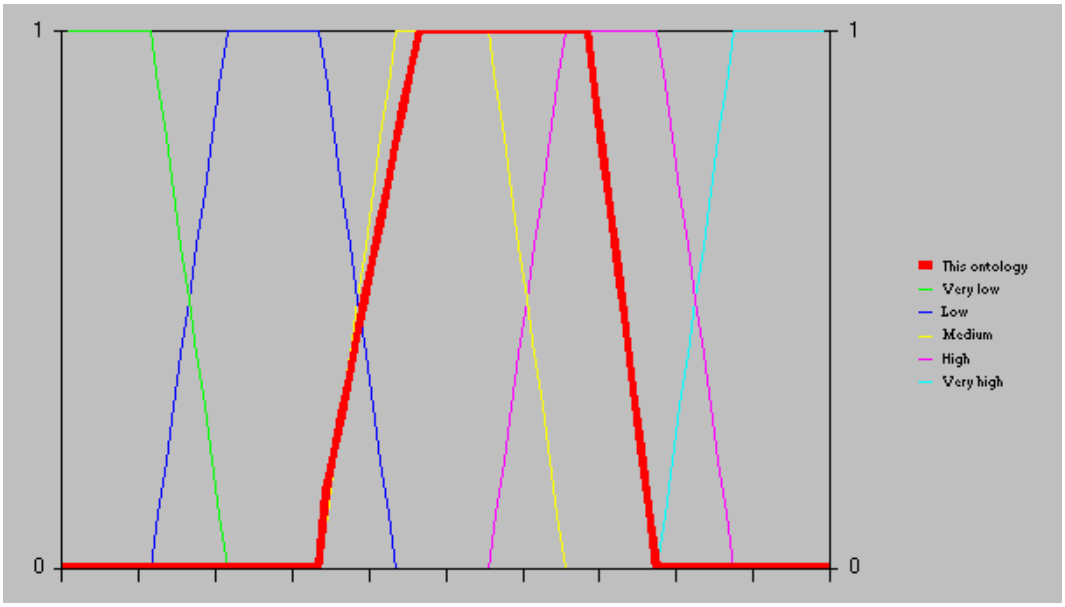


Figura 5.12. Gráfica de valoración de la ontología *Bibliographic_Data* del caso de estudio 5, respecto a una escala lingüística de 5 valores.

Ontología aconsejada: *Bibliographic-Data*

Los valores obtenidos, teniendo en cuenta todas las dimensiones, dan una medida mejor para la ontología *Bibliographic_Data* y con un valor bastante parecido para la ontología *Documents*, ya que ésta coincide en todas las dimensiones excepto en "contenido" con la ontología aconsejada, por estar implementada en el mismo lenguaje y poder utilizar los mismos entornos de desarrollo. Ninguna de ellas se conoce que haya sido creada usando ninguna metodología, y los costes de uso son nulos. La gráfica de la figura 5.12 muestra de forma individual la ontología *Bibliographic_Data* con una escala de comparación de cinco valores donde se observa que la ontología obtiene una medida entre media y alta.

5.4. CONCLUSIONES DE LA EVALUACIÓN DEL MÉTODO *ONTOMETRIC*

Utilizando los entornos software ROIS y OT, se han realizado 15 casos de prueba para evaluar las dimensiones: "lenguajes de implementación", "metodologías de desarrollo", "entornos software de desarrollo" y sobre el "contenido" de las ontologías. Además, se han realizado 15 casos de prueba donde intervienen todas las dimensiones de forma conjunta. En los anteriores apartados se muestra un ejemplo para cada uno de los casos. Se debe decir que, además de estos 75 casos de prueba realizados, se han hecho innumerables pruebas con las aplicaciones en las fases de desarrollo. En los 75 casos de prueba realizados, se han detectado los siguientes

errores:

- Un error en el valor de una característica de una instancia de un lenguaje de implementación. Debido a que se insertó erróneamente (de forma manual) este valor en la instancia.
- Un error en el valor de una característica de una instancia de un entorno software de desarrollo. Debido a que un valor de una característica de un entorno de desarrollo no fue actualizado convenientemente cuando el entorno adquirió esa cualidad.
- Dos errores en el valor de sendas características del contenido, de dos ontologías implementadas en *DAML+OIL*. Esto fue debido a que la aplicación *OntoWrapper_DAML* no extrajo convenientemente la información de dos ontologías implementadas en *DAML+OIL*, ya que no seguían completamente el formato estándar de la sintaxis de *DAML+OIL*.

Los valores obtenidos en los casos de prueba realizados, excepto en los cuatro mencionados, han sido conforme a los resultados esperados. El proceso de especificación de los valores en las características de forma multidimensional utilizando la aplicación ROIS es rápido e intuitivo. Las instancias obtenidas en las búsquedas multicaracterísticas y multidimensional han sido las esperadas. ROIS muestra toda la información de forma completa al usuario, y exporta los valores de las características de las instancias candidatas a la aplicación OT. Mediante la aplicación OT se permite, con una interfaz fácil de usar, indicar la importancia que tiene cada criterio con criterios "hermanos", y realiza los cálculos de las medidas de idoneidad tal y como se esperaba. Mediante la representación en forma de árbol del AMC, OT permite obtener una gráfica comparativa de idoneidad en cualquier nivel del AMC. OT, además, produce una detallada documentación de los pesos indicados por el usuario, de las ontologías candidatas analizadas, y de las medidas y gráficas comparativas obtenidas, mediante un informe impreso de resultados para la toma de decisiones.

Con las pruebas realizadas se puede afirmar que **es posible usar las aplicaciones que asisten al método *OntoMetric* y que, con los resultados obtenidos, se puede realizar una decisión justificada de elección de una ontología (o de una dimensión determinada) para usarla en un proyecto software**.

Capítulo 6

Conclusiones y Líneas Futuras

La principal aportación de este trabajo es la elaboración de una **métrica de valoración de idoneidad para ontologías que pueden ser usadas en un determinado sistema**. Se resuelve así el problema de la carencia de un método que sirva al usuario para decidir qué ontologías son las más adecuadas para incorporarlas en su proyecto. Para obtener estas mediciones, se ha elaborado un **método que valora cuantitativamente la idoneidad de las ontologías**, denominado ***OntoMetric***.

Las **principales innovaciones** de este trabajo son:

- Identificar los criterios que deben ser analizados para elegir una ontología. Estos criterios se han organizado en un marco multinivel de 160 características en el que el usuario puede escoger el grado de detalle que desee en los criterios, dependiendo de la exactitud requerida en el análisis. Estas características se han agrupado en las dimensiones de: lenguajes de implementación, metodologías de desarrollo, entornos software de desarrollo, costes de uso y el contenido de las ontologías. El marco multinivel de características ha sido validado mediante un cuestionario de opinión en el que varios usuarios de ontologías han revisado las características y su organización.
- Construcción de la ontología *Reference Ontology* (RO), siguiendo la metodología de desarrollo de ontologías *METHONTOLOGY*, y utilizando el entorno de desarrollo de ontologías *WebODE*. La RO contiene la información sobre las dimensiones: lenguajes de implementación, metodologías de desarrollo, entornos software de desarrollo, costes de uso y el contenido de las ontologías. Usando *WebODE* se han instanciado las dimensiones de la RO de "lenguajes", "metodologías", "entornos" y "costes". Se han creado aplicaciones software (*OntoWrappers*) que, de forma automática, extraen el contenido de las ontologías existentes, implementadas en sus respectivos lenguajes (*Ontolingua*, *DAML+OIL* y *X-WebODE*), y lo transforman en instancias válidas de la dimensión "contenido" de la RO.
- Construcción de una aplicación software (*Reference Ontology Instances Selector*) que permite seleccionar, desde la RO, las ontologías candidatas mediante consultas compuestas, sobre características en varias dimensiones del marco multinivel de

características.

- Adaptación del método AHP para asignar un valor cuantitativo de idoneidad a las ontologías candidatas obtenidas en el proceso anterior. El método AHP sirve para ayudar a la toma de decisiones multicriterio, y se ha adaptado a la reutilización de ontologías, conforme a los criterios especificados en el marco multinivel de características. Se ha modificado el proceso de valoración de AHP usando escalas lingüísticas, de forma que el usuario puede asignar un valor a las características de forma más intuitiva. Mediante los cuestionarios de opinión se ha obtenido un conjunto de pesos por defecto que el usuario puede tomar como referente, para hacer la valoración de los criterios del marco de características. Se ha desarrollado una aplicación software, denominada *OntoMetric Tool*, que: a) asiste al proceso de asignación de pesos de importancia sobre el marco multinivel de características, b) recoge de forma automática los valores de características de las ontologías candidatas suministradas por la aplicación *Reference Ontology Instances Selector*, c) realiza el cálculo de las medidas de idoneidad de las ontologías candidatas, y d) proporciona gráficas de comparación y documentación sobre los resultados obtenidos.

Para obtener los valores de la métrica, el usuario debe realizar dos procesos:

1. Proceso de **selección de ontologías**: se buscan las ontologías que cubran las necesidades y directrices del proyecto. Estas ontologías se denominan candidatas.
2. Proceso de **medición de idoneidad:** las ontologías candidatas encontradas se comparan con el fin de elegir las más idóneas respecto al sistema que pretende usarlas. Para realizar esta comparación se indica la importancia relativa que tiene cada uno de los criterios del marco multinivel de características. Por último se realiza el cálculo del valor de la métrica para cada ontología, que se obtiene sumando los valores que tiene la ontología en cada una de los criterios, ponderado por el peso de importancia asignado. Las medidas de idoneidad obtenidas para las ontologías candidatas sirven para elegir las ontologías más adecuadas para el proyecto.

Como resultado de la revisión bibliográfica realizada se puede decir que, hasta la fecha, la forma de decidir la ontología más adecuada para un proyecto estaba basada en la intuición y experiencia del usuario; ya que, **hasta ahora, no ha existido ningún método que asista al usuario en el proceso de elección de ontologías**. *OntoMetric* se ha desarrollado para cubrir esa carencia.

No es posible, por tanto, comparar *OntoMetric* con ningún otro método similar de elección de

ontologías para usarlas en un proyecto. Los casos de prueba realizados para la evaluación del método han mostrado que es aplicable y que, con las medidas obtenidas, se puede justificar la elección de una ontología o de una dimensión determinada.

Para resumir, se puede concluir que el trabajo realizado en esta memoria proporciona un **método teórico y un soporte tecnológico para ayudar al usuario a decidir sobre la elección de ontologías para incorporarlas a un sistema**. Este método está especialmente indicado para ser usado en proyectos de cierta importancia, donde una mala elección puede ocasionar importantes gastos, y en los casos en los que el usuario necesite justificar la decisión de elegir unas ontologías frente a otras.

No se puede considerar *OntoMetric* como un método revolucionario para la toma de decisiones, ya que la base teórica (AHP) en la que se sustenta ha sido aplicada a otros campos; pero sí es **original** el trabajo de adaptación al caso del uso de ontologías, del diseño de los procesos para aplicarlo y de las herramientas software que lo asisten.

Se debe indicar que la adaptación del método AHP con escalas lingüísticas de valoración se ha aplicado también en Ingeniería del Software a la reutilización de componentes software, y su resultado ha sido publicado en la conferencia *Development and Reengineering of Information Systems* en el congreso *Information System 2001 (IS2001)* celebrado en octubre del 2001 en Ljubljana, Slovenia [Loz01]. En este caso se utiliza, como marco de características, las identificadas para la reutilización de componentes software, cuyo resumen se publicó en el *Workshop on Component-based Software Engineering* dentro de la *22nd International Conference on Engineering (ICSE'00)* celebrado en Limerick, Irlanda, en junio de 2000 [Gom00].

La primera versión que se implementó de la *Reference Ontology* se construyó para servir como base de conocimientos de un sistema de búsquedas de ontologías denominado *(Onto)²Agent*. Este trabajo fue publicado por primera vez en el *Workshop on Applications of Ontologies and Problem-Solving Methods* dentro de la conferencia *European Conference on Artificial Intelligence (ECAI'98)* que tuvo lugar en Brighton, Reino Unido, en agosto de 1998 [Arp98], y una versión extendida en la revista *Knowledge and Information Systems, An International Journal, Springer-Verlag 2000* [Arp00a].

Las características identificadas en el marco multinivel de características de este trabajo se han tomado como punto de partida para construir la aplicación *OntoRoadmap* [37] del proyecto *OntoWeb* de la comunidad europea bajo el programa de "Tecnología de la Sociedad de la Información", IST-2000-29243.

[37] http://babage.dia.fi.upm.es/ontoweb/wp1/OntoRoadMap/index.html

Existen varias **líneas abiertas de investigación** que sería interesante abordar en futuros trabajos:

1. Analizar los pesos de importancia que se obtienen en el uso de *OntoMetric* en proyectos pertenecientes a un mismo escenario de aplicación. De esta forma, futuros usuarios que vayan a utilizar *OntoMetric* en un proyecto del mismo dominio podrían tener, como base de referencia, los pesos obtenidos en los diferentes escenarios.
2. Idear procesos y herramientas software que analicen, de forma automática, el contenido de las ontologías. En este momento, mediante la aplicación *Reference Ontology Instances Selector* se pueden buscar palabras clave en cualquier propiedad de los términos de las ontologías existentes; pero no está resuelto el problema de la búsqueda de términos sinónimos o búsqueda automática en las definiciones formales.
3. Utilizar el método *OntoMetric* para valorar la idoneidad de los conjuntos de instancias para nuevos sistemas. Para ello se debe establecer el esquema de características que deban ser analizadas para comparar diferentes conjuntos de instancias de un modelo conceptual, e idear procesos y aplicaciones para poder aplicar el método.
4. Se pueden realizar estudios sobre los valores de importancia que los usuarios asignan al marco multinivel de características en los diferentes proyectos, para deducir los aspectos más importantes que deben ser tenidos en cuenta a la hora de desarrollar nuevos lenguajes, metodologías y entornos de desarrollo de ontologías.

Bibliografía

Sobre Ingeniería del Conocimiento y Ontologías

[Agu98] Aguado G., Bateman J., Bañón A., Bernardos S., Fernández M., Gómez-Pérez A., Nieto. E, Olalla A., Plaza R. y Sanchez A. "ONTOGENERATION: Reusing Domain and Linguistic Ontologies for Spanish". *Workshop on Applications of Ontologies and PSMs. European Conference of Artificial Intelligence (ECAI'98). Págs. 1-10*. Brighton. Reino Unido. 1998.

[Arp98] Arpírez J., Gómez-Pérez A., Lozano-Tello A. y Pinto S. "(Onto)^{2}Agent: An Ontology-Based WWW Broker to Select Ontologies". *Workshop on Applications of Ontologies and PSMs. European Conference of Artificial* Intelligence *(ECAI'98). Págs. 16-24*. Brighton. Reino Unido. 1998

[Arp00a] Arpírez J., Gómez-Pérez A., Lozano-Tello A. y Pinto S. "Reference Ontology and (Onto)^{2}Agent: The Ontology Yellow Pages". *Knowledge and Information Systems, An International Journal. Editado por Springer-Verlag, vol.2 no.4, Págs. 387-412.* 2000.

[Arp00b] Arpírez J., "WebODE: Entorno de Desarrollo Distribuido de Ontologías para la Web". *Proyecto Fin de Carrera.* Facultad de Informática. Universidad Politécnica de Madrid. 2000.

[Arp01] Arpírez J., Corcho O., Fernández-López M. y Gómez-Pérez A. "WebODE: a Workbench for Ontological Engineering". *First International Conference on Knowledge Capture (K-CAP'01)*. Victoria B.C. Canada. 2001

[Bat90] Bateman J., Kasper T., Moore J. y Whitney R. "A General Organization of Knowledge for Natural Language Processing: The Penman Upper Model". *Technical Report, USC/ISI*. Marina del Rey, CA (USA), 1990.

[Bat94] Bateman J. "KPML: The KOMET-Penman (Multilingual) Development Environment". *Technical Report, GMD/IPSI*. Darmstadt. Germany. 1994.

[Bat95] Bateman J.A., Magnini B. y Fabris G. "The Generalized Upper Model Knowledge Base: Organization and Use". *Towards Very Large Knowledge Bases. IOS* Press. Págs. 60-72. 1995.

[Bec01] Bechhofer S. "OilEd University of Manchester". *Presentación en OntoWeb Workshop. Universal Ontology Editor Workshop*. Vrije Universiteit Amsterdam. The Netherlands. Febrero. 2001

[Ben98] Benjamins R., Decker S., Fensel D., Motta E., Plaza E., Schreiber G., Studer R. y Wielinga B. "IBROW3 An Intelligent Brokering Service for Knowledge-Component Reuse on the World-Wide Web". *Workshop on Applications of Ontologies and PSMs. (ECAI'98). Págs. 25-30*. Brighton, Reino Unido. 1998.

[Ben99] Benjamins R., Fensel D., Decker S. y Gómez-Pérez A. "(KA)^2: Building Ontologies for the Internet: a Mid-Term Report". *International Journal of Human-Computer Studies. Vol.51. Págs. 687-712.* 1999.

[Ber96] Bernaras A., Laresgoti I. y Corera J. "Building and Reusing Ontologies for Electrical Network Applications". *European Conference of Artificial Intelligence (ECAI'96). Editado por Wiley and Sons Ltd. Págs. 298-302.* Budapest. Hungary. 1996.

[Ber01] Berners-Lee T., Hendler J. y Lassila O. "The Semantic Web". *Scientific American. vol.284, no.5. Págs 34-43*. Mayo, 2001.

[Bor96] Borst P., Benjamins J., Wielinga B. y Akkermans H. "An Application of Ontology Construction". *Workshop on Ontological Engineering. European Conference of Artificial Intelligence (ECAI'96). Págs. 5-16*. Budapest. 1996.

[Bor97] Borst W. "Construction of Engineering Ontologies". *Tesis Doctoral*. University of Twente. Enschede. 1997.

[Bri99] Brickley D. y Guha R. "Resource Description Framework (RDF) Schema Specification". *W3C Proposed Recommendation*. Marzo, 1999. (http://www.w3.org/TR/PR-rdf-schema).

[Bro01] Brown M. y Flett A. "Enterprise-standard ontology environments". *Presentación en OntoWeb Workshop. Universal Ontology Editor Workshop.* Vrije Universiteit Amsterdam. The Netherlands. Febrero, 2001.

[Byl88] Bylander T. y Chandrasekaran B. "Generic Task in Knowledge-based Reasoning: The Right Level of *Abstraction for Knowledge Acquisition". Editores B. Gaines y J. Boose. Knowledge Acquisition for Knowledge Based Systems, vol. 1.Págs 65-77. Academic Press*. Londres. Reino Unido. 1988.

[Cha97] Chaudhri V.K., Farquhar A., Fikes R., Karp P.D. y Rice J.P. "The Generic Frame Protocol 2.0". *Technical Report. KSL-97-05.Standford University*. USA. Julio, 1997.

[Coa99] Coalition W.N. "Workflow Management Coalition". 1999. (http://www.aiim.org/wfmc/mainframe.htm).

[Cor00] Corcho O. y Gómez-Pérez A. "A Roadmap to Ontology Specification Languages". Proceedings of 11th European Workshop on Knowledge Acquisition. Modelling and Management (EKAW'00). Págs. 80-96. Jean-Les-Pins. Francia. 2000.

[Cor02] Corcho O. y Gómez-Pérez A. "Ontology Specification for the Semantic Web". *Pendiente de publicar en IEEE Intelligence System*. 2002.

[Dom98] Domingue J. "Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web". *Proceedings of the 11th Knowledge Acquisition Workshop. (KAW'98) Editado por B. Gaines and M. Musen*. Banff, Canada. Abril, 1998.

[Dui99] Duineveld A. J., Stoter R., Weiden M. R., Kenepa B. y Benjamins R. "Wondertools? A Comparative Study of Ontological Engineering Tools". *Proceedings of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop.(KAW'99).* University of Calgary/Stanford University. 1999.

[Duk95] Dukes-Schlossberg J., Mark W. y Kerber R. "Ontological Commitments and Domain-specific Architectures: Experience with Comet and Cosmos". *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing. Editado por N.J.I. Mars. IOS Press. Págs. 33-44.* Amsterdam. The NetherLands. 1995.

[Euz95] Euzenat J. "Building Consensual Knowledge Bases: Context and Architecture*". Towards Very Large Knowledge Bases. 2nd International Conference on Building and Sharing Very Large-scale Knowledge Bases (KBKS). Amsterdam IOS press. Págs. 143-155.* Enschede. The Netherlands. Abril, 1995.

[Far96] Farquhar A., Fikes R. y Rice J. "The Ontolingua Server: A Tool for Collaborative Ontology Construction". *Proceedings of the 10th Knowledge Acquisition for Knowledge-based Systems Workshop, vol.19, Págs. 1-44*. Banff. Alberta. Canada. 1996.

[Fen98] Fensel D., Decker S., Erdman M. y Studer R. "Ontobroker: The Very High Idea.". *Proceedings of the 11th International Flairs Conference (FLAIRS'98).* Sanibal Island. Florida. Mayo, 1998.

[Fen01] Fensel D. "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce". *Springer-Verlag.* Berlín, Alemania. 2001.

[Fer96] Fernández M. "CHEMICALS: Ontología para Elementos Químicos". *Proyecto Fin de Carrera.* Facultad de Informática. Universidad Politécnica de Madrid. Diciembre, 1996.

[Fer97] Fernández M., Gómez-Pérez A. y Juristo N. "METHONTOLOGY: From Ontological Art Toward Ontological Engineering". *Spring Symposium Series on Ontological Engineering. (AAAI'97). Págs. 33-40*. Stanford. California. USA. Marzo, 1997.

[Fer99a] Fernández M., Gómez-Pérez A., Pazos A. y Pazos J. "Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment". *IEEE Intelligent Systems, vol.14, no.1. Págs 37-46*. 1999.

[Fer99b] Fernández López M. "Overview of Methodologies for Building Ontologies". *Proceedings of the IJCAI'99. Workshop on Ontologies and PSMs.* Stockholm. Sweden. Agosto, 1999.

[Fer00] Fernández López M. "Método Bi-fase para la Conceptualización de Ontologías Basado en Meta-modelos". *Tesis Doctoral*. Facultad de Informática. Universidad Politécnica de Madrid. Octubre, 2000.

[Gen92] Genesereth M. y Fikes R. "Knowledge Interchange Format, version 3.0, Reference Manual". *Technical Report. Logic-92-86*. Computer Science Department. Stanford University. California. USA. 1992.

[Gom97] Gómez-Pérez A., Juristo N., Montes C. y Pazos J. "Ingeniería del Conocimiento". *Editado por Editorial Centro de Estudios Ramón Areces*. 1997.

[Gom98] Gómez-Pérez A. "Knowledge Sharing and Reuse". *The Handbook of Applied Expert Systems. Editado por J. Liebowitz, CRC Press*. Londres. Reino Unido. 1998.

[Gom99a] Gómez-Pérez A. y Benjamins R. "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods". *Proceedings of the IJCAI'99. Workshop on Ontologies and PSMs.* Stockholm. Sweden. Agosto, 1999.

[Gom99b] Gómez-Pérez A. y Benjamins R. "Applications of Ontologies and Problem-solving Methods". *AI Magazine vol.20, no.1. Págs. 119-122*. 1999.

[Gom99c] Gómez-Pérez A. "Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases". *Proceedings of the Knowledge Acquisition Workshop (KAW'99).* University of Calgary/Stanford University. 1999.

[Gom99d] Gómez-Pérez A. y Rojas-Amaya M. "Ontological Reengineering for Reuse". *Proceedings of the European Knowledge Acquisition Workshop (EKAW'99).* Dagstuhl Castle, Alemania. Mayo, 1999.

[Gru93a] Gruber T. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". *Technical Report KSL-93-04*. Knowledge Systems Laboratory. Stanford University. California. USA. 1993.

[Gru93b] Gruber T. "A Translation Approach to Portable Ontology Specifications". *Knowledge Acquisition, vol.5 no.2, Págs. 199-220.* 1993.

[Gru94] Gruber T. y Olsen R. "An Ontology for Engineering Mathematics". *Technical Report KSL-94-18*. Knowledge Systems Laboratory. Stanford University. California. USA. 1994.

[Gru95a] Gruber T. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". *International Journal of Human-Computer Studies, 43. Págs. 907-928.* 1995.

[Gru95b] Gruninger M. y Fox M. "Methodology for the Design and Evaluation of Ontologies". *Proceedings of the IJCAI'95. Workshop on Basic Ontological Issues in Knowledge Sharing.* Montreal. Canada. 1995.

[Gua95] Guarino N. y Giaretta P. "Ontologies and Knowledge Bases: Towards a Terminological Clarification" *Toward Very Large Knowledge Bases. Knowledge Building & Knowledge Sharing. Editado por IOS Press. Págs. 25-32.* 1995.

[Gua98] Guarino N. "Some Ontological Principles for Design Upper Lexical Resources". *Proceedings of the 1st International Conference on Language Resources and Evaluation*. *Págs.527-534.* Granada. España. 1998.

[Gua00] Guarino N. y Welty C. "A Formal Ontology of Properties". *Proceedings of the ECAI'00 Workshop on Applications of Ontologies and PSMs.* Berlin. Agosto, 2000.

[Hef99] Heflin J., Hendler J. y Luke S. "Applying Ontology on the Web: A Case Study". *Workshop Putting Intelligence in the Web. 5th International Work-Conference on Artificial and Natural Neural Networks (IWANN'99).* Págs. 715-724. Alicante. España. 1999.

[Hef00] Heflin J. y Hendler J. "Searching the Web with SHOE". *Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01. Editado por AAAI Press. Págs. 35-40.* Menlo Park. California. USA. 2000

[Hor00] Horroks I., Fensel D., Harmelen F., Decker S., Erdmann M. y Klein M. "OIL in a Nutshell". *Workshop on Applications of Ontologies and PSMs. European Conference of Artificial Intelligence (ECAI'00). Págs. 73-77*. Berlin. Germany. Agosto, 2000.

[Hor01] Horrocks I. y van Harmelen F. "Reference Description of DAML+OIL Ontology Markup Language". *Draft Report.* 2001.

[Hov97] Hovy E. "What Would it Mean to Measure an Ontology?". *Internal Paper. ISI of the University of Southern California*. Marina del Rey. USA. 1997.

[Hov98] Hovy E. "Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses". *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC'98).* Granada. España. 1998

[Hum93] Humphreys B. y Lindberg D. "The UMLS Project: Making the Conceptual Conection between Users and the Information They Need". *Bulletin of the Medical Library Association 81(2). Págs. 170-183.* 1993.

[Kar99] Karp R., Chaundhri V. y Thomere J. "XOL: An XML-Based Ontology Exchange Language". *Technical Report. Artificial Intelligence Center.* SRI International. Agosto, 1999.

[Ken98] Kent R. "Conceptual Knowledge Markup Language (version 0.2)". 1998 (http://wave.eecs.wsu.edu/CKRMI/CKML.html).

[Kif95] Kifer M., Lausen G. y Wu J. "Logical Foundations of Object-Oriented and Frame-Based Languages". *Journal of the ACM, vol.42, no.4. Págs. 741-843.* Julio, 1995.

[Kni94] Knight K. y Luk S. "Building a Large Knowledge Base for Machine Translation". *American Association for Artificial Intelligence. (AAAI). Págs 773-778.* Seattle. USA. 1994.

[Las99] Lassila O. y Swick R. "Resource Description Framework (RDF). Model and Syntax Specification". *W3C Recommendations*. Enero, 1999. (http://www.w3.org/TR/PR-rdf-syntax).

[Len90] Lenat D. y Guha R.V. "Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project". *Editado por Addison-Wesley Publishing Company, Inc.* California. USA. 1990.

[Luk00] Luke S. y Heflin J. "SHOE 1.01. Proposed Specification". *SHOE Project*. Febrero 2000. (http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm).

[Mae01] Maedche A. y Staab S. "Comparing Ontologies. Similarity Measures and a Comparative Study". *Internal Report no.408. Institute AIFB. University of Karlsruhe*. Karlsruhe. Germany. March 2001.

[Mah96] Mahesh K. "Ontology Development for Machine Translation: Ideology and Methodology". *Technical Report. Computer Research Laboratory.* New Mexico State University. 1996.

[MGr91] McGregor R. "Inside the LOOM classifier". *SIGART bulletin #2(3). Págs 70-76.* Junio, 1991.

[MGu00] McGuinness D., Fikes R., Rice J. y Wilder S. "The Chimaera Ontology Environment." *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI'00).* Austin. Texas. Agosto, 2000.

[MEn99] McEntire R. et al. "An Evaluation of Ontology Exchange Languages for Bioinformatics". *Technical Report.* Agosto, 1999.

[Mil90] Miller G. "WORDNET: an Online Lexical Database". *International Journal of Lexicography, 3(4). Págs. 235-312*. 1990.

[Miz93] Mizoguchi R. "Knowledge Acquisition and Ontology". *Proceedings of the KB&KS'93. Págs, 121-128.* Tokyo. 1993.

[Miz95] Mizoguchi R., Vanwelkenhuysen J. e Ikeda M. "Toward Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing". *IOS Press. Págs. 46-59*. 1995.

[Mot95] Motta E. "KBS Modelling in OCML". *Proceedings of the 5th Workshop on Modelling Languages for KBS. Vrije Universiteit*. Amsterndam. Enero, 1995.

[Mot99] Motta E. "Reusable Components for Knowledge Modelling". *IOS Press.* Amsterdam, 1999.

[Nec91] Neches R., Fikes R., Finin T., Gruber T., Patil R., Senator T. y Swartout W. "Enabling Technology for Knowledge Sharing". *AI Magazine. Págs. 36-56*. Otoño, 1991.

[Nil01] Niles I. y Pease A. "Origin of the IEEE Standard Upper Ontology". *Proceedings of the IJCAI'01. Workshop on the IEEE Standard Upper Ontology*. Seattle. Washington. USA. Agosto, 2001.

[Noy97] Noy N.F. y Hafner C. "The State of the Art in Ontology Design". *AI MAGAZINE. Págs 53-74.* Otoño 1997.

[Noy01] Noy N.F. et al. "Creating Semantic Web Contents with Protege-2000". *IEEE Intelligent Systems 16(2). Págs. 60-71.* 2001.

[Pin99] Pinto S., Gómez-Pérez A. y Martins J. "Some Issues on Ontology Integration". *Proceedings of the IJCAI'99. Workshop on Ontologies and PSMs.* Stockholm. Sweden. Agosto, 1999.

[Pin00] Pinto S. y Martins J. "Reusing ontologies". *AAAI 2000 Spring Symposium on Bringing Knowledge to Business Processes. Págs. 77-84.* 2000.

[Pre99] Preece A., Hui K. y Gray P. "KRAFT: Supporting Virtual Organizations Throught Knowledge Fusion". Architecture for Knowledge Fusion and Tranformation". *Artificial Intelligence for Electronic Commerce: Papers from the AAAI-99 Workshop. Editado por T Finin & B Grosof. AAAI Press. Págs. 33-38.* 1999.

[Ros94] Rosner D. "Generating Multilingual Documents from a Knowledge Base: The TechDoc Project". *Technical Report, FAW Ulm.* Ulm. Germany. 1994.

[Sch95] Schreiber A., Wielinga B. y Jansweijer W. "The KACTUS View on the 'O' Word". *Proceedings of the IJCAI'95. Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal. Canada. 1995.

[Sch99] Scheiber G., Akkermans H. y Anjewierden A. "Knowledge Engineering and Management. The CommonKADS Methodology". *MIT Press*. Massachussets. 1999.

[Sow97] Sowa J. "Knowledge Representation: Logical, Philosophical, and Computational Foundations". *Brooks Cole Publishing Co., Pacific Grove, CA. USA.* 1997.

[Sta00] Staab S. y Maedche A. "Ontology Engineering Beyond the Modeling of Concepts and Relations". *Proceedings of the ECAI'00 Workshop on Applications of Ontologies and PSMs.* Berlin. Agosto, 2000.

[Sta01] Staab S. y Maedche A. "Knowledge Portals --- Ontologies at Work". *AI Magazine, vol.21, no.2.* Verano, 2001.

[Ste00] Steve G., Gangemi A. y Pisanelli D. "Ontological Analysis and Integration of Terminologies: Towards An Environmental Reference Ontology Library". *Presentado en el Open Forum on Metadata Registries .* Santa Fe. New México. Enero, 2000.

[Sto93] Stock O., Carenini G., Cecconi F., Franconi E., Lavelli A., Magnini B., Pianesi F., Ponzi M., Samek V. y Strapparava C. "Alfresco: Enjoying the Combinational of Natural Processing and Hypermedia for Information Exploration". *Intelligent Multimedia Interfaces, c9. Editado por Mark Maybury. The MIT Press. Págs. 197-224*. 1993.

[Stu98] Studer S., Benjamins R. y Fensel D. "Knowledge Engineering: Principles and Methods". Data and Knowledge Engineering, 25, 161-197, 1998.

[Swa97] Swartout B., Patil R., Knight K. y Russ T. "Towards Distributed Use of Large-Scale Ontologies". *AAAI'97 Spring Symposium Series on Ontological Engineering. Págs. 138-148.* 1997.

[Usc95] Uschold M. y King M. "Towards a Methodology for Building Ontologies". *Proceedings of the IJCAI'95. Workshop on Basic Ontological Issues in Knowledge Sharing.* Montreal. Canada. 1995

[Usc96a] Uschold M. "Building Ontologies: Towards a Unified Methodology". *Proceedings of Expert Systems (ES'96). Cambridge*. Reino Unido. Diciembre, 1996.

[Usc96b] Uschold M. y Grüninger M. "Ontologies: Principles, Methods and Applications". *Knowledge Engineering Review, vol.2, Págs. 93-155.* Junio, 1996.

[Usc98a] Uschold M. "Where are the Killer Applications?". *Workshop on Applications of Ontologies and PSMs. (ECAI'98). Págs. 107-111.* Brighton, Reino Unido. 1998.

[Usc98b] Uschold M., King M., Moralee S. y Zorgios Y. "The Enterprise Ontology The Knowledge Engineering Review". *Special Issue on Putting Ontologies to Use. vol.13. Editado por Mike Uschold and Austin Tate.* 1998.

[Usc99] Uschold M. y Jasper R. "A Framework for Understanding and Classifying Ontology Applications" *Proceedings of the IJCAI'99. Workshop on Ontologies and PSMs.* Stockholm. Sweden. Agosto, 1999.

[vHa99] van Harmelet F. y Fensel D. "Surveying Notations for Machine-Processable Semantics of Web Sources". *Proceedings of the IJCAI'99. Workshop on Ontologies and PSMs.* Stockholm. Sweden. Agosto, 1999.

[vHe97] van Heist G., Schreiber A. y Wielinga B. "Using Explicit Ontologies in KBS Development". *International Journal of Human-Computer Studies, vol.45. Págs. 183-292.* 1997.

[vVe94] van der Vet P., de Jong H., Mars N., Speel P. y Stal W. "Plinius Intermediate Report". *Memoranda Informatica. Págs. 94-35.* University of Twente. Enschede. The Netherlands. 1994.

[Wat86] Waterman D. "A Guide to Expert Systems". *Addison-Wesley.* Massachusets. USA. 1986.

[Wei99] Weinstein P., William P. y Birmingham W. "Comparing Concepts in Differentiated Ontologies". *Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99).* Banff. Alberta. Canada. Octubre 1999.

[Wie92] Wielinga B., Schreiber A. y Breuker J. "KADS: A Modeling Approach to Knowledge Engineering". *Knowledge Acquisition (4). Págs. 5-53*. 1992.

[Zyl00a] van Zyl J. y Corbett D. "Population of a Framework for Understanding and Classifying Ontology Applications". *Proceedings of the ECAI'00. Workshop on Applications of Ontologies and PSMs.* Berlin. Agosto, 2000.

[Zyl00b] van Zyl J. y Corbett D. "A Framework for Comparing the Use of a Linguistic Ontology in an Application". *Proceedings of the ECAI'00. Workshop on Applications of Ontologies and PSMs.* Berlin. Agosto, 2000.

Sobre Ingeniería del Software y Métricas

[Abt98] Abts C., Clark B., Devnani-Chulani S., Horowitz E., Madachy E., Reifer D., Selby R. y Steece B. "COCOMO II Model Definition Manual". *Technical Report. Center for Software Engineering, USC.* 1998.

[Agr92] Agresti W. y Evanco W. "Projectin Software Defects in Analyzing Ada Designs". *IEEE Transactions. Software Engineering. vol.18, no.11. Págs. 988-997*. 1992.

[Alb79] Albrecht A. "Measuring Application Development". *Proceedings of IBM Applications Development Joint SHARE/GUIDE Symposium. Págs 83-92.* California. USA. 1979.

[Ant96] Anthes G. "Software Reuse Plans Bring Paybacks". *Computerworld, vol.27, no.49. Págs. 73-76.* 1996.

[Bar91] Barnes B. y Bollinger T. "Making Software Reuse Cost Effective". *IEEE Software, vol.1. Págs. 13-24*. 1991.

[Bas90] Basili V., Rombach H. y Delis A. "Ada Reusability and Measurement". *Computer Science Technical Report Series*. University of Maryland. Mayo, 1990.

[Bas94] Basili V., Briand L. y Thomas W. "Domain Analysis for the Reuse of Software Development Experiences". *Proceedings of 19th Annual Software Engineering Workshop, NASA/GSFC.* Greenbelt. MD. Diciembre, 1994.

[Bie93] Bieman J. y Karuranithi S. "Candidate Reuse Metrics for Object-Oriented and Ada Software". *Proceedings of IEEE-CS 1st International Software Metrics Symposium*. 1993.

[Boe81] Boehm B. "Software Engineering Economics". *Prentice-Hall. Englewood Cliffs. New Jersey.* 1981.

[Bol90] Bollinger T. y Pfleeger S. "The Economics of Reuse: Issues and Alternatives". *Proceedings of the 8th Annual National Conference on Ada Technology. Págs. 436-447.* Atlanta 1990.

[CSI00] Consejo Superior de Informática, Ministerio para las Administraciones Públicas. "Metodología de Planificación y Desarrollo de Sistemas de Información. Métrica v3.0". *MAP*. España. 2000.

[Dav91] Davis T. "The Reuse Capability Model: A Basic for Improving an Organization's Reuse Capability". *Proceedings of the 2nd International Workshop on Software Engineering. IEEE Computer Society Press, , Págs. 329-338.* Austin TX. USA. 1991.

[Dem89] Demming E. "Out of Crisis". *MIT Centre for Advanced Engineering Study.* Cambridge. MA, USA. 1989.

[Fen95] Fenton N., Iizuka Y. y Whitty R. "Software Quality Assurance and Measurement: A Worldwide Perspective". *International Thomson Computer Press.* Londres. Reino Unido. 1995.

[Fen96] Fenton N. "Software Metrics, A Rigorous Approach". *2nd Edition. Editado por Chapman & Hall.* Londres. Reino Unido. 199 6.

[Fer83] Ferrari D. "Computer System Performance Evaluation". *Prentice Hall, Englewood Cliffs.* NJ. USA. 1983.

[Fra87] Frakes W. y Nejmeh B. "Software Reuse thought Information Retrieval". *Proceedings of the 20th Hawaii International Conference on System Sciences. Editado por B. Shriver, Kailua-Kona.* Hawai. 1987.

[Fra90] Frakes W. y Gandel P. "Representing Reusable Software". *Software Technology, vol.32, no.10. Págs. 653-664.* 1990.

[Fra94a] Frakes W. y Terry C. "Reuse Level Metrics". *Proceedings of the 3rd International Conference on Software Reuse. IEEE Computer Science Press. Págs. 139-148.* CA. USA. 1994.

[Fra94b] Frakes W. y Pole T. "An Empirical Study of Representation Methods for Reusable Components". *IEEE Transactions Software Engineering, vol.20, no.8. Págs. 617-630*. 1994.

[Fra95] Frakes W. y Fox C. "Modeling Reuse across the Software Lifecycle". *Journal of Systems Software vol.30, no.3. Págs. 295-301.* 1995.

[Fra96a] Frakes W. y Terry C. "Software Reuse: Metrics and Models". *ACM Computing Surveys, vol.28, no.2, Págs. 415-435.* Junio, 1996.

[Fra96b] Frakes W. y Fox C. "Quality Improvement Using a Software Reuse Failure Modes Model". *IEEE Transactions. Software Engineering. vol.24, no.4. Págs. 274-279*. Abril, 1996.

[Gaf89] Gaffney J. y Durek T. "Software Reuse - Key to Enhanced Productivity: Some Quantitative Models". *Information and Software Technology, vol.31, no.5. Págs. 258-267.* Junio, 1989.

[Gil88] Gilb T. "Principles of Software Engineering Management". *Addison-Wesley, Reading.* MA. USA. 1988.

[Gom00] Gómez-Pérez A. y Lozano-Tello A. "Impact of Software Components Characteristics above Decision-making Factors". *Workshop on Component-based Software Engineering. 22nd International Conference on Engineering (ICSE'00). Págs. 15-23*. Limerick, Ireland. Junio, 2000.

[Hal77] Halstead M. "Elements of Software Science". *Elsevier.* North Holland. 1977.

[Har92] Harrison W. "An Entropy-based Measure of Software Complexity". *IEEE Transactions on Software Engineering. vol.18, no.11. Págs. 1025-1029.* 1992.

[IEE96] IEEE Standard for Developing Software Life Cycle Processes. *IEEE Computer Society.* NY. USA. Abril, 1996.

[ISO91] International Standard ISO/IEC 9126. "Information technology, Software product evaluation, Quality characteristics and guidelines for their use". *International Organization for Standardization, International Electrotechnical Commission.* Geneva, 1991.

[Jel72] Jelinsky Z. y Moranda P. "Software Reliability Research". Statistical Computer Performance Evaluation., Academic Press. Págs. 465-484. NY. USA. 1972.

[Jon85] Jones C. "A Process-Integrated Approach to Defect Prevention". *IBM Systems Journal, vol.4, no.2, Págs. 150-187.* 1985.

[Jon86] Jones C. "Programmer Productivity". *McGraw-Hill.* NY. USA. 1986.

[Kan87] Kang K. "A Reuse-Based Software Development Methodology". *Proceedings of the Workshop on Software Reuse. Editado por G. Booch, SEI, MCC, Software Productivity Consortium.* Boulder. Colorado. USA. 1987.

[Kar95] Karlsson E. "Software Reuse: A Holistic Approach". *Wiley*. NY. USA. 1995.

[Kit89] Kitchenham B. y Walker J. "A quantitative approach to monitoring software development". *Software Engineering Journal, vol. 4, no.1, Págs. 2-13.* 1989.

[Kle75] Kleinrock L. "Queueing Systems". *Vol.1: Theory y Vol.2: Computer Applications, Wiley*. NY. USA. 1975.

[Kol91] Koltun P. y Hudson A. "A Reuse Maturity Model". *4th Annual Workshop on Software Reuse.* Herndon VA. USA. 1991.

[Lit73] Littlewood B. y Verrall J. "A Bayesian Reliability Growth Model for Computer Software". *Journal of the Royal Statistical Society, vol.22, Págs. 332-344*. 1973.

[Loz01] Lozano-Tello A. y Gómez-Pérez A. "Application of the AHP in the Election of Software Components". *Development and Reingeneering of Information Systems in Information System 2001 (IS'01)*. Ljubljana. Slovenia. Octubre, 2001.

[MCa76] McCabe T. "A Software Complexity Measure". *IEEE Transactions on Software Engineering vol.2, no.4, Págs. 308-320.* 1976.

[MCa77] McCall J., Richard P. y Walters G. "Factors in Software Quality". *NTIS AD-A049-014, 015, 055.* Noviembre, 1977.

[Mus75] Musa J. "A Theory of Software Reliability and its Applications". *IEEE Transactions on Software Engineering, vol.1, no.3. Págs. 312-327.* 1975.

[NIS94] National Institute of Standards and Technology "Glossary of Software Reuse Terms". Diciembre. 1994.

[Pia95] Piattini M. "Métricas para el Desarrollo Orientado a Objetos". *Revista Base Informática, vol.26, Págs. 18-22.* 1995.

[Pou93] Poulin J., Caruso J. y Hancock D. "The Business Case for Software Reuse". *IBM Systems Journal, vol.32, no.4, Págs. 567-594.* 1993.

[Pou97] Poulin J.S. "Measuring Software Reuse: Principles, Practices, and Economic Models". *Addison Wesley Longman Inc*. 1997.

[Pre97] Pressman R.S. "Software Engineering: A Practitioner's Approach". *McGraw-Hill, Inc*. 1997.

[Pri87] Prieto-Díaz R. y Freeman P. "Classifying *Software* for Reusability". *IEEE Software , vol.4, no.1, Págs. 6-16.* Enero, 1987.

[Put78] Putman L. "A General Empirical Solution to the Macrosoftware Sizing and Estimating Problem". *IEEE Transaction on Software Engineering, vol.4, no.4, Págs. 345-61.* 1978.

[RAE99] Real Academia Española. "Diccionario de la Lengua Española". *21ª edición. Espasa Calpe*, Madrid 1999.

[Rio89] Ríos S. "Procesos de Decisión Multicriterio". *Editorial Eudema, Manuales.* 1989.

[Rio96] Ríos J. y Serras J. "Métricas para OO: Estado Actual, Herramientas y Tendencias". *Revista NOVÁTICA, vol.121, Págs. 25-36*. 1996.

[Roy90] Roy B. "Decision Aid and Decision Making". *European Journal of Operation Research, vol.45, Págs. 324-331.* 1990.

[Rou95] Rout T. "SPICE: A Framework for Software Process Assessment". *Software Process - Improvement and Practice, vol.1, no.2, Págs. 57-66*. Agosto, 1995.

[Saa77] Saaty T. "A Scaling Method for Priorities in Hierarchal Structures". J*ournal of Mathematical Psychology, vol.15, Págs. 234-281* 1977.

[Saa90] Saaty T. "How to Make a Decision: The Analythic Hierarchy Process". E*uropean Journal of Operational Research, vol.48, Págs. 9-26.* 1990.

[Vin92] Vincke P. "Multicriteria Decision Aids". *Wiley*. New York. USA. 1992.

Anexo I

Cuestionario de Opinión sobre Características para Seleccionar Ontologías

En este anexo se muestra un cuestionario que propone 160 preguntas sobre la importancia de características de ontologías. Las características sondeadas en este cuestionario se identifican con las que se indican en el marco multinivel de características. El cuestionario ha sido rellenado por 10 usuarios familiarizados con el desarrollo de ontologías.

El cuestionario ha servido, en primer lugar, para validar la conveniencia de las características identificadas y su disposición en los diferentes apartados. Además, ha servido para obtener los pesos de importancia asignados al árbol multinivel de características, que podrán ser utilizados por los usuarios de *OntoMetric* como ponderación por defecto en los procesos de medición. Los resultados del cuestionario se muestran en el anexo II de esta memoria.

Mapa del cuestionario

C- Sobre el contenido.
- C1) Sobre la coincidencia de los conceptos (clases).
- C2) Sobre las relaciones.
- C3) Sobre la taxonomía de los conceptos.
- C4) Sobre los axiomas.

M- Sobre la metodología que se ha seguido para construir la ontología.
- M1) Sobre la precisión de la metodología.
- M2) Sobre la usabilidad de la metodología.
- M3) Sobre la madurez de la metodología.

T- Sobre los entornos software de desarrollo de la ontología.
- T1) Sobre las prestaciones de los entornos.
- T2) Sobre la visualización de los términos de la ontología.
- T3) Sobre la edición de los términos de la ontología.
- T4) Sobre aspectos de interacción con la ontología.
- T5) Sobre aspectos metodológicos.
- T6) Sobre aspectos cooperativos.
- T7) Sobre la traducción a-desde otros lenguajes.
- T8) Sobre integración de ontologías.

L- Sobre el lenguaje en el que está implementada.
- **L1) Sobre la representación.**
  - L11) Respecto a los conceptos (clases).
  - L12) Respecto a los atributos de conceptos.
  - L13) Respecto a las facetas de los atributos.
  - L14) Respecto a la representación de relaciones.
  - L15) Respecto a la representación de taxonomías.
  - L16) Respecto a la representación de axiomas.
  - L17) Respecto a la representación de reglas de producción.
- **L2) Sobre el mecanismo de inferencia.**
  - L21) Respecto al potencial de razonamiento del lenguaje.
  - L22) Respecto al motor de inferencias asociado al lenguaje de implementación.

Cost- Sobre los costes de uso de la ontología.

CUESTIONARIO SOBRE CARACTERÍSTICAS PARA SELECCIONAR ONTOLOGÍAS

Entrevistado: _____

El supuesto es construir un sistema y buscar una ontología existente para integrarla en él. ¿En qué debemos fijarnos?, ¿qué es lo importante?, ¿de qué va a depender elegir una ontología frente a otras? Por favor, contesta a este cuestionario evitando pensar en un proyecto concreto, sino considerando las características que a ti te parezcan importantes para seleccionar una ontología para cualquier tipo de sistemas. Vamos a suponer también que existe un gran número de ontologías que pueden ser reutilizadas.

Valora el cuestionario sobre características de ontologías indicando qué importancia das a cada una de ellas. Para ello, marca una o varias casillas, e indica tu opinión sobre la valoración realizada. Si no estás seguro de lo que significa la pregunta, preferimos que no contestes. Si piensas que olvidamos o no es necesaria alguna característica te agradeceríamos que nos lo comentaras.

SOBRE EL CONTENIDO

Se valora aquí la importancia que tiene la coincidencia de los términos que están representados en la ontología, con los términos que se supone debe utilizar el nuevo sistema. Es decir, sobre la parte del dominio que la ontología cubre de las necesidades de nuestro nuevo sistema.

En general, sobre el grado de coincidencia del contenido de la ontología con las necesidades del sistema:

- ☐ Es suficiente que la ontología tenga un dominio parecido al buscado. Servirá como base para reelaborar una ontología para el sistema.
- ☐ La ontología debe coincidir en parte con las necesidades del sistema. Se deberá hacer un proceso de adaptación.
- ☐ La ontología debe coincidir en un alto porcentaje con las necesidades del sistema. Las adaptaciones deberían ser mínimas.
- ☐ La ontología debe coincidir casi totalmente con las necesidades del sistema. Las adaptaciones deberían ser sólo puntuales.
- ☐ La ontología debe coincidir totalmente con lo buscado. No debería hacerse ninguna adaptación.
- ☐ No estoy seguro.
- ☐ Otra idea:

SOBRE LA COINCIDENCIA DE LOS CONCEPTOS (CLASES)

Se evalúa en este apartado la coincidencia de los conceptos existentes en la ontología con las necesidades del sistema que se va a desarrollar.

C1 En **general**, di si te parece importante que coincidan los conceptos en la ontología evaluada con lo necesitado en el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C11 Los conceptos que son esenciales para el sistema se encuentran en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C12 Los conceptos que son esenciales para el sistema se encuentran en los niveles superiores de la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C13 Los conceptos se encuentran descritos convenientemente en lenguaje natural.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C14 La especificación formal de los conceptos coincide con la descripción de los conceptos en lenguaje natural.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C15 Los atributos de los conceptos los describen de forma precisa.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C16 El número de conceptos que son representados en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LAS RELACIONES

Se evalúa en este apartado la coincidencia de las relaciones entre conceptos existentes en la ontología con las necesidades del sistema que se va a desarrollar.

C2 En **general**, di si te parece importante que coincidan las relaciones definidas en la ontología con lo necesitado en el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C21 Las relaciones que son esenciales para el sistema se encuentren definidas en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C22 Los conceptos de la ontología se encuentran relacionados tal y como se necesitan para el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C23 La especificación formal de las relaciones coincide con la descripción de las relaciones en lenguaje natural.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C24 Las relaciones tienen especificada la aridad tal y como se necesita para el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C25 Las relaciones tienen especificadas propiedades formales necesarias para el sistema: (reflexividad, irreflexividad, simetría, asimetría, antisimetría, transitividad, intransitividad).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C26 El número de relaciones definidas en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LA TAXONOMÍA DE LOS CONCEPTOS

Sobre cómo están clasificados los conceptos en la ontología y su coincidencia con la organización necesaria para el sistema.

C3 En **general**, di si te parece importante la coincidencia de las taxonomías de conceptos con las que necesita el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C31 Los conceptos de la ontología se encuentran clasificados desde varias perspectivas.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C32 Existen especificadas relaciones del tipo "Not-Subclass-Of" necesarias para el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C33 Existen especificadas relaciones de partición disjuntas necesarias para el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C34 Existen especificadas relaciones de partición exhaustivas necesarias para el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C35 La profundidad máxima a la que llega la jerarquía de conceptos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C36 El número medio de subclases que tienen las clases.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LOS AXIOMAS

La existencia de axiomas en la ontología quizá sea necesaria en el nuevo sistema para hacer deducciones, mantener la consistencia o restringir el valor de las instancias.

C4 En **general**, di si te parece importante que existan axiomas.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C41 Los axiomas pueden utilizarse para hacer deducciones, utilizados para resolver consultas.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C42 Existen axiomas para completar los valores sobre los atributos de conceptos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C43 Existen axiomas que se utilizan para verificar la consistencia de términos en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C44 Existen axiomas definidos como elementos independientes (no vinculados a conceptos).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

C45 El número de axiomas definidos en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LA METODOLOGÍA QUE SE HA SEGUIDO PARA CONSTRUIR LA ONTOLOGÍA

La metodología seguida para desarrollar la ontología puede establecer la forma de incorporar nuevos términos para un nuevo sistema o para modificarlos. Además, una ontología desarrollada con una metodología adecuada tendrá mayores garantías de estar construida correctamente. Se valora en este apartado tanto la idoneidad del proceso para llevar a cabo el desarrollo de la ontología, como la calidad del producto que se le supone a una ontología que siga la metodología. Se debe tener en cuenta que, a menudo, los entornos software de desarrollo de ontologías siguen una determinada metodología. Por favor, intenta evaluar la metodología por separado, ya que existe un apartado para evaluar la importancia de los entornos.

En general, sobre la metodología que se ha empleado para desarrollar la ontología:
☐ La metodología no me parece una característica importante para seleccionar una ontología.
☐ La metodología no me parece una característica fundamental para seleccionar una ontología.
☐ La metodología me parece importante para seleccionar una ontología.
☐ La metodología me parece muy importante para seleccionar una ontología.
☐ La metodología me parece fundamental para seleccionar una ontología.
☐ No estoy seguro.
☐ Otra idea:

SOBRE LA PRECISIÓN DE LA METODOLOGÍA

Una metodología que no describa con exactitud las actividades de desarrollo, las técnicas empleadas y el personal involucrado en cada una de las fases puede dificultar el proceso de incorporar o modificar términos en la ontología, o incluso puede llevar a crear modelos conceptuales incorrectos.

M1 En **general**, di si te parece importante la precisión de la metodología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M11 La metodología tiene bien delimitadas las fases de desarrollo de ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M12 La metodología tiene bien especificadas las actividades que deben realizarse en cada fase.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M13 La metodología tiene bien identificado el personal que interviene en cada fase y su función.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M14 La metodología tiene bien especificadas las técnicas que deben emplearse en cada fase.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M15 La metodología tiene bien especificados los productos que deben obtenerse tras la ejecución de cada fase.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LA USABILIDAD DE LA METODOLOGÍA

Si la metodología describe engorrosos procesos de desarrollo, no existe información suficiente o ésta es poco clara, la modificación de términos de la ontología puede ser un proceso lento y complejo.

M2 En **general**, di si te parece importante la usabilidad de la metodología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M21 La descripción de las actividades y técnicas que deben realizarse en cada fase se entiende fácilmente.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M22 Existe suficiente material de consulta para aprender a usar la metodología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M23 Existen ejemplos completos de aplicación de la metodología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LA MADUREZ DE LA METODOLOGÍA

Una metodología que se ha empleado para construir ontologías usadas en proyectos reales, en diferentes dominios, y para el desarrollo de ontologías de cierta envergadura, dará mayores garantías de que las ontologías estén construidas de forma correcta.

M3 En **general**, di si te parece importante la madurez de la metodología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M31 El número de ontologías desarrolladas para proyectos reales siguiendo la metodología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M32 La metodología ha sido empleada en desarrollar ontologías de diversos dominios.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

M33 La metodología ha sido empleada en desarrollar ontologías de cierta envergadura.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LOS ENTORNOS SOFTWARE DE DESARROLLO DE ONTOLOGÍAS

Se valoran en este apartado las características de las aplicaciones software que pueden visualizar el contenido de las ontologías, editar sus términos, realizar uniones e integraciones de varias ontologías, etc., ya que quizás deban realizarse adaptaciones para las necesidades del proyecto que se va a desarrollar. Por favor, intenta valorar los entornos software sin juzgar los lenguajes de implementación que soportan el conocimiento de las ontologías, ya que este proceso se realiza en el siguiente apartado.

En general, sobre los entornos existentes asociados a la ontología:

- ☐ Los entornos software no me parecen importantes para seleccionar una ontología.
- ☐ Los entornos software no me parecen fundamentales para seleccionar una ontología.
- ☐ Los entornos software me parecen importante para seleccionar una ontología.
- ☐ Los entornos software me parecen muy importantes para seleccionar una ontología.
- ☐ Los entornos software me parecen fundamentales para seleccionar una ontología.
- ☐ No estoy seguro.
- ☐ Otra idea:

SOBRE LAS PRESTACIONES DE LOS ENTORNOS

La calidad y posibilidades de los entornos de desarrollo de ontologías pueden colaborar en un mejor entendimiento de su contenido y en acelerar el proceso de adaptación de éstas al sistema.

T1 En **general**, di si te parecen importantes las prestaciones que en conjunto ofrecen los entornos asociados a la ontología..
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T11 Los entornos pueden instalarse localmente.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T12 Los entornos pueden instalarse en una red local.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T13 Los entornos funcionan en Internet mediante un navegador.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T14 La claridad que ofrece la interfaz (en cuanto a la organización de las opciones, uniformidad, ausencia de ambigüedades en las opciones, ayudas contextuales, etc.)
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

.

T15 La rapidez de los procesos para visualizar o editar términos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

.

T16 Los entornos están suficientemente probados y parecen estar libres de fallos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LA VISUALIZACIÓN DE LOS TÉRMINOS DE LA ONTOLOGÍA

Sobre cómo nos muestra los términos y su contenido las aplicaciones software de visualización de ontologías.

T2 En **general**, di si te parece importante cómo los entornos nos muestran el contenido de los términos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T21 Permiten visualizar toda la información que se encuentre especificada en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T22 Permiten ver la información con el nivel de detalle deseado.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T23 Permiten ver, de alguna forma, la taxonomía de los conceptos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T24 Permiten usar el modo gráfico para mostrar las relaciones *ad-hoc*.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LA EDICIÓN DE LOS TÉRMINOS DE LA ONTOLOGÍA

En el caso de tener que hacer modificaciones en los términos de la ontología para adaptarla al nuevo sistema, deben valorarse ciertas características de las aplicaciones que ayudarán a realizar estas tareas. Debe tenerse en cuenta que los entornos pueden estar limitados por las capacidades expresivas del lenguaje que soporta la ontología.

T3 En **general**, di si te parece importante la forma en la que los entornos permiten modificar los términos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T31 Con el entorno podemos implementar todo lo que podemos implementar directamente con el lenguaje de implementación.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T32 Permite realizar modificaciones de los términos en cualquier momento; es decir, siempre podemos editar la ontología y añadir nuevos términos, modificar y borrar términos, sus propiedades, sus relaciones, etc.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T33 Permite realizar modificaciones en las relaciones taxonómicas de forma gráfica.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T34 Permite trabajar de forma gráfica con relaciones *ad-hoc*.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE ASPECTOS DE INTERACCIÓN CON LA ONTOLOGÍA

El acceso a ontologías puede hacerse de forma remota a un servidor web, o se puede realizar un proceso de exportación y explotar esta información localmente. Por otro lado, se consumirá mucho tiempo y esfuerzo si para acceder o modificar términos de la ontología desde el nuevo sistema se deben implementar nuevos módulos software que realicen estas tareas. Algunos entornos proporcionan sus interfaces para ser utilizadas en otros proyectos.

T4 En **general**, sobre las facilidades de interacción que tiene la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T41 La ontología puede ser exportada desde el servidor y se puede utilizar localmente; no es necesario acceder a sus términos de forma remota.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T42 La ontología proporciona interfaces adecuadas de acceso a los términos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T43 Existe documentación suficiente para saber utilizar las primitivas de las interfaces de acceso.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T44 Las interfaces de acceso a los términos son fácilmente adaptables; el código es abierto y puede ser modificado.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T45 Existe documentación suficiente para saber modificar las primitivas de las interfaces de acceso.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE ASPECTOS METODOLÓGICOS

Para la creación de ontologías, algunos entornos software sirven de soporte tecnológico que cubren parcial o totalmente el ciclo de vida de alguna metodología de desarrollo de ontologías.

T5 En **general**, di si te parece importante que los entornos se fundamenten en los procesos teóricos de alguna metodología de desarrollo de ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T51 Los entornos cubren todas las actividades del ciclo de vida del desarrollo de ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T52 Los entornos cubren las actividades más importantes (o adecuados para hacerlos asistidos por software) del ciclo de vida del desarrollo de ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T53 Los entornos proporcionan documentación suficiente sobre los productos creados o modificados en las fases de desarrollo de las ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T54 Los entornos realizan comprobaciones de verificación para los cambios realizados en los términos de la ontología. Al realizar modificaciones comprueban la consistencia de la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE ASPECTOS COOPERATIVOS

Es usual que los proyectos sean desarrollados por varias entidades, quizás en diferentes localizaciones. Los entornos de desarrollo de ontologías pueden ofrecer algunas prestaciones para trabajar de esta forma.

T6 En **general**, di si te parece importante que el entorno esté diseñado para trabajar con varias personas en la misma ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T61 Los entornos permiten crear grupos de trabajo para editar una misma ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T62 Los entornos permiten la edición simultánea de la misma ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T63 Los entornos realizan el bloqueo de la ontología a otros usuarios cuando se está editando.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T64 Los entornos realizan el bloqueo de los términos concretos editados rechazando modificaciones a otros usuarios.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T65 Los entornos avisan a usuarios conectados de los cambios producidos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T66 Los entornos identifican, de alguna forma, los cambios que ha hecho cada usuario.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LA TRADUCCIÓN A-DESDE OTROS LENGUAJES

Para el nuevo sistema, se deben tener en cuenta los lenguajes en los que podemos implementar la ontología. A la hora de valorar el lenguaje en el que está implementada la ontología (en el siguiente apartado), deberemos considerar la evaluación del que más nos interese, si existen traductores que pueden realizar esta tarea.

T7 En **general**, di si te parece importante la existencia de traductores de lenguajes de ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T71 Los entornos permiten importar ontologías desde otros lenguajes de implementación "clásicos" de ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T72 Los entornos permiten importar ontologías desde lenguajes de marcado *web*.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T73 Los entornos permiten exportar ontologías a otros lenguajes de implementación "clásicos" de ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T74 Los entornos permiten exportar ontologías a lenguajes de marcado *web*.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T75 Los traductores consiguen que la ontología en el lenguaje objeto pierda lo menos posible de la semántica que tenía en el lenguaje fuente. Lo "menos posible" significa que la pérdida es ocasionada por la menor capacidad expresiva del lenguaje objeto.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T76 Los entornos permiten supervisar las traducciones a otros lenguajes.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE INTEGRACIÓN DE ONTOLOGÍAS

Algunos entornos posibilitan o favorecen la integración de varias ontologías existentes en una destino. De esta forma se pueden reutilizar los términos de ontologías desde librerías de ontologías. Otros entornos permiten mezclar (*merge*) los términos definidos en varias ontologías de manera que se forma una ontología destino consistente con las definiciones de las ontologías mezcladas.

T8 En **general**, que el entorno permita integrar ontologías existentes.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T81 La facilidad en realizar el proceso de integración.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T82 La forma de referenciar los términos integrados se hace fácilmente, sin ninguna notación especial.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T83 El entorno tiene la capacidad de integrar sólo los términos que se deseen desde una ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T84 El entorno realiza verificaciones al integrar o mezclar varias ontologías, de forma que mantiene la consistencia.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T85 El entorno ofrece asistencia para mezclar (*merge*) de forma manual las taxonomías de dos ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

T86 El entorno ofrece algún tipo de asistencia para unir de forma semi-automática las taxonomías de varias ontologías.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE EL LENGUAJE EN EL QUE ESTÁ IMPLEMENTADA

Las ontologías se implementan en uno o varios lenguajes de implementación. Para nuestro sistema quizá necesitemos que el lenguaje nos ofrezca una serie de prestaciones, expresivas y de razonamiento. Indica qué te parece importante que deba tener el lenguaje. Ten en cuenta que sólo estamos valorando el lenguaje de implementación de la ontología; por favor, no consideres en este apartado las posibilidades y facilidades que pueden ofrecer los entornos software asociados al lenguaje, que han sido valoradas en el apartado anterior.

En general, sobre el lenguaje de implementación:
☐ El lenguaje no me parece una característica importante para seleccionar una ontología.
☐ El lenguaje no me parece una característica fundamental para seleccionar una ontología.
☐ El lenguaje me parece importante para seleccionar una ontología.
☐ El lenguaje me parece muy importante para seleccionar una ontología.
☐ El lenguaje me parece fundamental para seleccionar una ontología.
☐ No estoy seguro.
☐ Otra idea:

SOBRE LA REPRESENTACIÓN DEL CONOCIMIENTO

El lenguaje de implementación va a dar unas capacidades expresivas que quizá necesitemos para explotar ese conocimiento, para añadir nuevos términos a la ontología existente, o para adaptarla a las necesidades de nuestro sistema.

L1 En **general** sobre la capacidad de representación del conocimiento que ofrece el lenguaje:
☐ La capacidad de representación no me parece una característica importante para seleccionar una ontología.
☐ La capacidad de representación no me parece una característica fundamental para seleccionar una ontología.
☐ La capacidad de representación me parece importante para escoger una ontología.
☐ La capacidad de representación me parece muy importante para escoger una ontología.
☐ La capacidad de representación me parece fundamental para escoger una ontología.
☐ No estoy seguro.
☐ Otra idea:

Respecto a la representación de conceptos (clases), instancias, hechos y afirmaciones

L11 En **general**, di si te parece importante la forma en la que se pueden representar los conceptos en la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L111 Poder definir instancias de conceptos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L112 Poder usar metaclases (crear clases como instancias de otras clases existentes).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L113 Poder definir clases sin usar metaclases existentes.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L114 Poder aseverar hechos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L115 Poder declarar afirmaciones.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Respecto a los atributos de conceptos

L12 En **general**, di si te parece importante para la representación del conocimiento los tipos de atributos que pueden ser utilizados en el lenguaje.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L121 Poder utilizar atributos de clase (con un valor igual para todas las instancias del concepto).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L122 Poder utilizar atributos de instancia (definidos en una clase pero que pueden tomar valores diferentes para cada instancia).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L123 Poder definir atributos locales de conceptos (que afectan a un único concepto).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L124 Poder definir atributos globales (no vinculados directamente a conceptos específicos)
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L125 Poder definir atributos polimórficos (que pueden tener distinto comportamiento en distintos conceptos).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L126 Poder redefinir atributos de clases superiores, para representar excepciones en la herencia.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Respecto a las facetas de los atributos

L13 En **general**, di si te parece importante para la representación las posibilidades que ofrecen las facetas de atributos en el lenguaje.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L131 Poder utilizar facetas para definir valores por defecto para los atributos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L132 Poder utilizar facetas para definir el tipo de valor para los atributos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L133 Poder utilizar facetas para definir el número mínimo y máximo de valores que puede tomar el atributo.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L134 Poder utilizar facetas para definir conocimiento procedural asociado a acciones.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L135 Poder añadir nuevas facetas, diferentes a las ya existentes.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Respecto a la representación de relaciones

L14 En **general**, di si te parece importante las capacidades de representación de las relaciones entre conceptos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L141 La existencia de relaciones tipo función (que devuelvan un argumento).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L142 Poder utilizar relaciones de aridad mayor que binaria.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L143 Poder definir nuevas relaciones ad-hoc.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L144 Poder restringir el tipo de los argumentos que se pasan a las relaciones.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L145 Poder restringir la integridad en el valor de los argumentos que se pasan a las relaciones.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L146 Que las relaciones permitan asociar definiciones operacionales para inferir valores de atributos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L147 Permite declarar propiedades matemáticas en las relaciones.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Respecto a la representación de taxonomías

L15 En **general**, di si te parece importante las posibilidades en la definición de taxonomías que ofrece el lenguaje.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L151 La existencia de la relación entre conceptos "Subclase de" (o alguna similar).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L152 La existencia de la relación entre conceptos "No subclase de" (o alguna similar).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L153 Poder realizar descomposiciones exhaustivas en clases disjuntas de una clase padre.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L154 Poder realizar descomposiciones en clases disjuntas de una clase padre.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L155 Permite que una clase sea subclase de varias clases padre.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L156 Permite que una instancia sea instancia de varias clases.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Respecto a la representación de axiomas

L16 En **general**, di si te parece importante que pueden definirse axiomas en el lenguaje.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L161 Poder representar axiomas vinculados a conceptos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L162 Poder representar axiomas no vinculados a conceptos (axiomas con nombre).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L163 Poder representar axiomas con lógica de primer orden.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L164 Poder representar axiomas con lógica de orden mayor que el primero.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Respecto a la representación de reglas de producción

L17 En **general**, di si te parece importante las capacidades de representación de reglas de producción.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L171 Poder definir premisas de forma conjuntiva.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L172 Poder definir premisas de forma disyuntiva.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L173 Poder asignar a la regla la prioridad de aplicación.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L174 Poder asignar a la regla valores de verdad.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L175 Poder asignar a la regla el tipo de encadenamiento.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L176 Se pueden asociar procedimientos a las reglas de producción que modifiquen información en la base de conocimientos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE EL MECANISMO DE INFERENCIA

Dependiendo del lenguaje de implementación, y condicionado por la capacidad de representar del conocimiento que posea, se va a tener un mayor o menor potencial de razonamiento. Quizá este potencial de razonamiento es utilizado por algún motor de inferencia existente en el lenguaje de implementación.

L2 En **general**, sobre el mecanismo de inferencia (por el potencial de razonamiento del lenguaje y por la existencia de un motor de inferencia) que ofrece el lenguaje, creo que:
☐ El mecanismo de inferencia no me parece una característica importante para seleccionar una ontología.
☐ El mecanismo de inferencia no me parece una característica fundamental para seleccionar una ontología.
☐ El mecanismo de inferencia me parece importante para escoger una ontología.
☐ El mecanismo de inferencia me parece muy importante para escoger una ontología.
☐ El mecanismo de inferencia me parece fundamental para escoger una ontología.
☐ No estoy seguro.
☐ Otra idea:

Respecto al potencial de razonamiento del lenguaje

L21 En **general**, di si te parece importante el potencial de razonamiento que ofrece el lenguaje.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L211 Permite herencia múltiple.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L212 Permite razonamiento monótono.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L213 Permite razonamiento no monótono.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L214 Utiliza las redefiniciones de atributos de clases superiores para realizar excepciones en la herencia.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L215 Los axiomas se pueden utilizar para mantener la consistencia del sistema mediante comprobación de restricciones.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L216 Los procedimientos que se definen en el lenguaje se pueden ejecutar.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L217 Las reglas de producción tienen asociado algún mecanismo de encadenamiento (hacia delante, hacia atrás, o ambos).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Respecto al motor de inferencias asociado al lenguaje de implementación

L22 En **general**, di si te parece importante que exista un motor de inferencia ya implementado que realiza razonamiento con los conocimientos de la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L221 La completitud y corrección del motor de inferencia implementado.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L222 La capacidad del motor de inferencia para realizar clasificaciones automáticas.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L223 El motor de inferencia considera, en los procesos de razonamientos, las excepciones en los atributos.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L224 El motor de inferencia maneja la herencia múltiple para realizar razonamiento.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

L225 Poder construir un motor de inferencia nuevo (además del existente).
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

SOBRE LOS COSTES DE USO DE LA ONTOLOGÍA

En este apartado se examinarán las características que llevan a estimar los costes asociados a incorporar la ontología al sistema. Este aspecto puede ser clave cuando el proyecto tenga un presupuesto ajustado, o cuando los costes de adquisición y explotación sean elevados.

En **general**, sobre los costes que ocasionen usar la ontología
- ☐ El coste estimado no me parece un factor importante para seleccionar una ontología.
- ☐ El coste estimado no me parece factor fundamental para seleccionar una ontología.
- ☐ El coste estimado me parece importante para escoger una ontología.
- ☐ El coste estimado me parece muy importante para escoger una ontología.
- ☐ El coste estimado me parece fundamental para escoger una ontología.
- ☐ No estoy seguro.
- ☐ Otra idea:

Cost1 El precio de adquisición de la ontología, para explotarla en el sistema.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Cost2 Los costes que van a ocasionar adquirir los recursos hardware y software para poder utilizar la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Cost3 El precio de adquisición de las interfaces software de acceso a la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Cost4 El precio de adquisición de los entornos software asociados a la ontología.
☐ No es importante. ☐ No es fundamental. ☐ Es importante. ☐ Es muy importante. ☐ Es fundamental.
☐ Comentario sobre la valoración:

Por favor, dinos qué características hemos olvidado. Agradeceremos cualquier sugerencia.

Anexo II

Descripción del Funcionamiento de los OntoWrappers

En este anexo se describe cómo los *OntoWrappers* extraen la información relevante de las ontologías existentes y la transforman en instancias de la dimensión "contenido" de la RO. Para hacer esto, se han construido *OntoWrappers* que extraen la información de las ontologías implementadas en los lenguajes *Ontolingua*, *DAML+OIL* y *X-WebODE*, por ser estos los lenguajes en los que están representadas las ontologías disponibles en los servidores de ontologías: *Ontolingua Server*, en el servidor *WebODE*, en las página *web* de *SHOE*, del *Ontology Group at ITBM-CNR* y de *DAML Ontology Library*. Los *OntoWrappers* generan la información en ficheros en formato *X-WebODE*, que pueden ser importados como conjuntos de instancias de la dimensión "contenido" de la RO desde la plataforma *WebODE*. En este anexo se describe, de forma muy general, en el apartado II.1 la sintaxis de *X-WebODE*. En los apartados II.2, II.3 y II.4 se explica el funcionamiento de los *OntoWrappers*: *Ontolingua_OntoWrappers*, *DAML_OntoWrappers* y *X-WebODE_OntoWrappers*, respectivamente.

II.1. Breve Descripción de la Sintaxis de *X-WebODE*

X-WebODE es el formato que utiliza el entorno de desarrollo de ontologías *WebODE* para exportar e importar el modelo conceptual y los conjuntos de instancias de ontologías a/desde ficheros *XML*. El lenguaje *X-WebODE* posee un vocabulario propio de representación incorporando significado a distintas etiquetas de *XML*. La *URL* de la *DTD* usada por *X-WebODE* es `http://babage.dia.fi.upm.es/webode/DTD/webode_1_0.dtd`. Una descripción detallada de la sintaxis de *X-WebODE* aparece en [Arp00b]. En este apartado se comenta el significado de algunas etiquetas elementales y para poder seguir los ejemplos.

La etiqueta raíz se denomina <Ontology> y dentro de la delimitación de su ámbito [38] aparecen en primer lugar las etiquetas <Name>, <Author> y <Creation-Date> que referencian al nombre, autor y fecha de creación de la ontología, respectivamente. Dentro de la etiqueta <Ontology>

[38] En XML, una etiqueta *A* delimita su **ámbito** desde que aparece la forma *<A>* hasta que termina con *</A>*. Dentro de este ámbito, se pueden anidar otras etiquetas con la misma sintaxis.

aparecen dos apartados: <Conceptualization> para indicar que, a continuación, el código *XML* define el modelo conceptual de la ontología; e <Instances> que indica que dentro de sus delimitadores de ámbito se van a definir los conjuntos de instancias de la ontología. Cada conjunto de instancias se va a delimitar con la etiqueta <Instance-Set>, y cada instancia concreta dentro de un conjunto se delimita con la etiqueta <Instance>.

Para el modelo conceptual de la RO, las únicas etiquetas que se van a usar son las relativas a conceptos (<Concept>) y a relaciones (<Term-Relation>), puesto que la RO no contiene funciones, axiomas, ni términos importados de otras ontologías. Todos los atributos usados en la RO son atributos de instancia, que se denotan con <Instance-Attribute>. En ellos se debe definir su nombre, descripción, tipo de dato permitido, y cardinalidad mínima y máxima, en las correspondientes etiquetas internas a <Instance-Attribute> denotadas con <Name>, <Description>, <Type>, <Minimum-Cardinality>, y <Maximum-Cardinality>. En la figura II.1 se muestra un ejemplo de cómo se implementa el concepto *Class* definido en la dimensión "contenido" de la RO en la sintaxis *XML* de *WebODE* (*X-WebODE*), y cómo se define la instancia *Automobile*, siguiendo el modelo conceptual de la RO, en el mismo formato *X-WebODE*.

```
<Concept>
 <Name>Class</Name>

 <Instance-Attribute>
 <Name>Class_Name</Name>
 <Description>Name of the class</Description>
 <Type>String</Type>
 <Minimum-Cardinality>1</Minimum-Cardinality>
 <Maximum-Cardinality>1</Maximum-Cardinality>
 </Instance-Attribute>

 <Instance-Attribute>
 <Name>Class_Description</Name>
 <Description>Short description of the class</Description
 <Type>String</Type>
 <Minimum-Cardinality>0</Minimum-Cardinality>
 <Maximum-Cardinality>1</Maximum-Cardinality>
 </Instance-Attribute>
 <Instance-Attribute>

 <Name>Formal_Description_Of_Class</Name>
 <Description>Formal description of the class,
             literal specification</Description>
 <Type>String</Type>
 <Minimum-Cardinality>0</Minimum-Cardinality>
 <Maximum-Cardinality>1</Maximum-Cardinality>
 </Instance-Attribute>

</Concept>
```

Descripción del modelo conceptual de la RO

Figura II.1. Ejemplo de definición en *X-WebODE* del concepto *Class*, y ejemplo de definición de la clase *Automobile* **(1/2)**.

```
<Instance>
 <Name>Automobile</Name>
 <Instance-of>Class </Instance-of>
 <Class>
  <Name>Class</Name>
  <Attribute>
      <Name>Class_Name</Name>
      <Value>Automobile</Value>
  </Attribute>
  <Attribute>
      <Name>Class_Description</Name>
      <Value>Any old sort of car.</Value>
  </Attribute>
  <Attribute>
      <Name>Formal_Description_Of_Class</Name>
      <Value>(Define-Class Automobile (?X)
             "Any old sort of car."
             :Def
             (And (Wheeled-Vehicle ?X)))
      </Value>
  </Attribute>
 </Class>
</Instance>
```

Descripción de las instancias de la RO

Figura II.1. Ejemplo de definición en *X-WebODE* del concepto *Class*, y ejemplo de definición de la clase *Automobile* **(2/2)**.

II.2. Extracción del "Contenido" desde Ontologías Implementadas en *Ontolingua*

Se ha desarrollado una aplicación software, denominada ***Ontolingua_OntoWrapper***, que extrae la información relevante desde ficheros de texto con ontologías implementadas en el lenguaje *Ontolingua*. Dicha aplicación genera ficheros *XML* en formato *X-WebODE*, compatible con el modelo conceptual de la dimensión "contenido" de la *RO*. De esta forma se puede incorporar información sobre las ontologías de *Ontolingua* como instancias de la RO en el entorno de desarrollo de ontologías *WebODE*, utilizando su módulo de importación. Para desarrollar esta aplicación, se ha analizado la sintaxis de *Ontolingua* y se han construido los patrones sintácticos para convertir la información relevante de los ficheros de *Ontolingua* en instancias de la dimensión "contenido" de la *RO*.

II.2.1. El lenguaje *Ontolingua*

El lenguaje "tradicional" más utilizado para implementar ontologías es *Ontolingua*. Está basado en el lenguaje *KIF* [Gen92] y en la *Frame Ontology* [Far96]; con este lenguaje se pueden representar clases, relaciones, funciones, instancias y axiomas. Estos elementos son los que va a detectar *Ontolingua_OntoWrapper* y los va a transformar en instancias válidas del modelo conceptual de la RO en la dimensión "contenido". *Ontolingua* permite representar ontologías utilizando exclusivamente el vocabulario de la *Frame Ontology* o utilizando expresiones en *KIF*.

La sintaxis completa del lenguaje *Ontolingua* se encuentra en la *URL* http://WWW-KSL-SVC.stanford.edu:5915/doc/ontolingua/reference-manual/index.html. Los formatos de instrucciones que aparecen en este apartado se muestran como ejemplo para describir cómo el *Ontolingua_OntoWrapper* extrae información de las ontologías representadas en *Ontolingua*.

Para representar los términos (clases, relaciones, funciones, instancias y axiomas) de las ontologías, *Ontolingua* usa una sintaxis similar para el formato basado en la *Frame Ontology*, usando la palabra clave **Define-Frame.** Un ejemplo de cómo definir clases en *Ontolingua* aparece en la figura II.2.

La sintaxis para definir instrucciones en el formato basado en *KIF*, a grandes rasgos, se puede definir como: un paréntesis abierto, seguido de una palabra clave que identifica el término que se va a definir (**Define-Class**, **Define-Instance**, **Define-Relation**, **Define-Function** o **Define-Axiom**), seguido del nombre que se asigna al término, una serie de elementos que identifican qué se está representando y finaliza en un paréntesis cerrado.

Algunas de las palabras clave, que van a ser tenidas en cuenta por el analizador del *Ontolingua_OntoWrapper*, son:

:def indica que lo especificado a continuación son condiciones necesarias para las instancias que pertenezcan a la clase definida.

:iff-def indica que lo especificado a continuación son condiciones necesarias y suficientes para las instancias que pertenezcan a la clase definida.

:class-slots indica que lo especificado a continuación son atributos de clase.

:instance-slots indica que lo especificado a continuación son atributos de instancia.

Una **clase** en Ontolingua se representa en alguno de los siguientes formatos:

Formato1:

```
(Define-Class nombre_de_la_clase (?variable)
      ["descripción en LN de la clase"]
      {:def | iff-def }sentencia_KIF
      (AND  [(nombre_de_superclase  ?variable )]+  )
      [:class-slots (nombre_del_atributo_de_clase)]
      [:instance-slots ( nombre_del_atributo_de_clase)]
      [otras_sentencias_KIF]*
)
```

} Formato para describir clases en *Ontolingua*

Figura II.2. Formato de definición en *Ontolingua* de una clase, y dos ejemplos de definición de clases **(1/2)**.

Formato2:

```
(Define-Frame nombre_de_la_clase
      :Own-Slots
      ([(Documentation "descripción en LN de la clase")]
      (Instance-Of Class [superclase]+ )
      [(Subclass-Of superclase )]* ]
      [otras_sentencias_KIF]* )
      :Template-Slots
      ( [(nombre_de_atributo (sentencia_KIF))]+ )
      [otras_sentencias_KIF]*
)
```

Formato para describir clases en *Ontolingua*

```
(Define-Class Automobile (?X)
      "Any old sort of car"
      :def
      (AND  (Wheeled-Vehicle ?X) )
)

(Define-Frame Vehicle-For-Sale
      :Own-Slots
      ((Documentation "A vehicle that we are trying to sel
      (Instance-Of Class)
      (Subclass-Of Vehicles Product))
      :Template-Slots
      (List-Price (Minimum-Cardinality 0))
)
```

Dos ejemplos de definición de clases en *Ontolingua*

Figura II.2. Formato de definición en *Ontolingua* de una clase, y dos ejemplos de definición de clases **(2/2)**.

II.2.2. Patrones sintácticos de transformación para la extracción de información desde *Ontolingua*

Para completar los valores de las características de "contenido" de la RO se han identificado los patrones sintácticos de transformación a partir de la sintaxis de *Ontolingua*. Como ejemplo, a continuación se muestran en la figura II.3.a y II.3.b los patrones sintácticos para detectar la especificación de una clase (con la notación Define-class y Define-frame), que completaría la característica *Class* en la RO; en la figura II.3.c y II.3.d se muestra cómo se detectaría la descripción en lenguaje natural (también conocido con el nombre de atributo "documentación") de un concepto (con la notación Define-class y Define-frame), que completaría la característica *Class_Description* en la RO. Se ha seguido la notación de autómatas finitos para expresar estos patrones sintácticos. El *Ontolingua_OntoWrapper* se ha implementado siguiendo la especificación de los patrones sintácticos definidos.

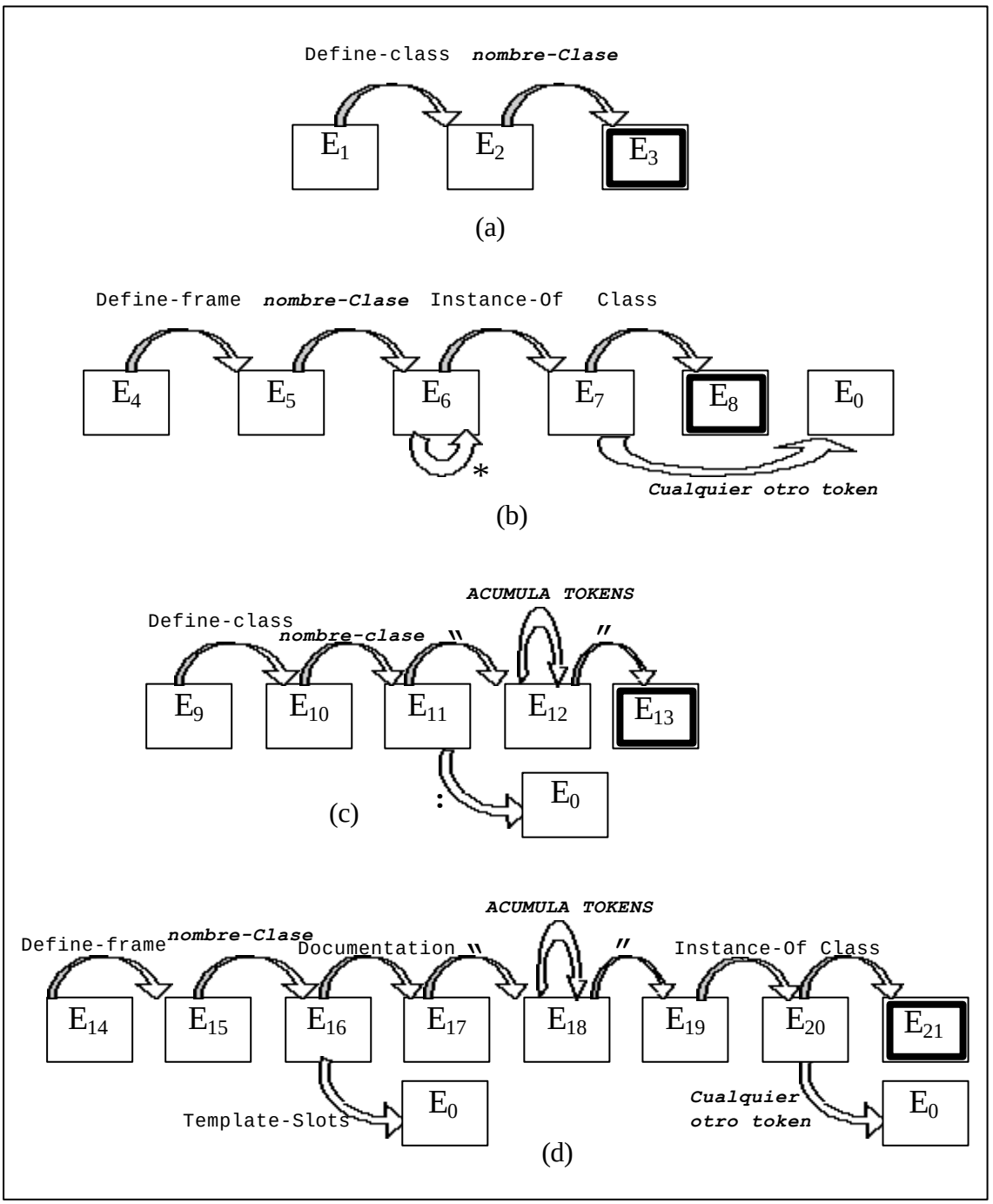


Figura II.3. Patrones sintácticos para detectar: (a) una clase definida con Define-Class; (b) una clase definida con Define-Frame; (c) la descripción de una clase definida con Define-Class; y (d) la descripción de una clase definida con Define-Frame.

II.2.3. Funcionamiento de la aplicación *Ontolingua_OntoWrapper* y ejemplo de detección de tokens

A partir de la definición de los patrones sintácticos se ha implementado la aplicación *Ontolingua_OntoWrapper*, que tomará como entrada un fichero de texto en formato *ASCII* con la ontología en el lenguaje *Ontolingua*, y creará, como salida, un fichero *XML* en formato *X-*

WebODE, que puede ser importado por el entorno *WebODE* como instancia del modelo conceptual de la dimensión "contenido" de la RO. La aplicación puede considerarse como un conjunto de *parsers* que implementan los patrones sintácticos definidos, y un generador de ficheros XML en el formato mencionado.

Como ejemplo muy simple de detección de tokens, a continuación se muestra el proceso de detección de la clase *Automobile* y de su descripción en lenguaje natural, incluida en la ontología *Vehicles*. Se usarán los patrones sintácticos representados en la figura II.3.

Detección de clases: Usando los patrones de transformación de la figura II.3.a y II.3.c, el analizador sintáctico recorre el fichero de la ontología implementada en *Ontolingua* buscando términos de tipo "clase". En una estructura de lista se almacenan los tokens de este tipo encontrados hasta ese momento, como se muestra en la figura II.4.

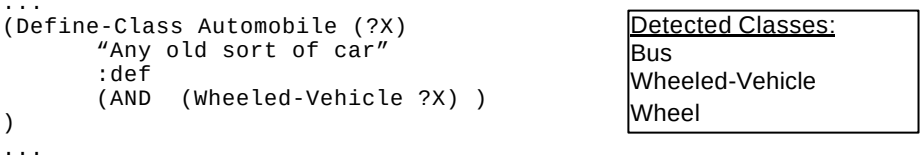


Figura II.4. Ejemplo de representación de una clase en *Ontolingua*, y estructura de tipo lista con las clases detectadas hasta el momento por el *Ontolingua_Ontowrapper*.

En este proceso, utilizando concretamente el patrón de transformación de la figura II.3.a, se parte del estado E_1, y tras detectar el token `Define-Class` almacena el siguiente token en la lista de conceptos "clase" encontrados, como aparece en la figura II.5.

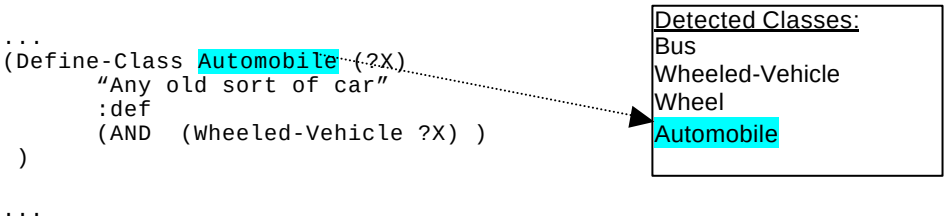


Figura II.5. Inserción en la estructura de tipo lista, de la clase detectada por el *Ontolingua_Ontowrapper*.

Detección de la descripción de una clase: usando el patrón de transformación de la figura II.3.c, el *parser* correspondiente recorre el fichero en *Ontolingua* buscando la descripción en lenguaje natural de la clase identificada. Al igual que en el proceso anterior, el elemento identificado se almacena en una estructura de lista y se vincula con el token tipo "clase" relacionado, como se muestra en la figura II.6.

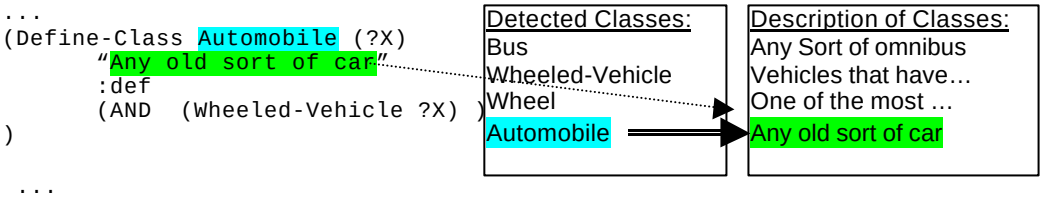


Figura II.6. Inserción en la estructura de tipo lista, de la descripción de la clase detectada por el *Ontolingua_Ontowrapper*.

II.2.4. Ejemplo de generación del fichero en X-WebODE

A partir de los elementos detectados por los *parsers* se genera el fichero de salida en *X-WebODE*. En el ejemplo de la figura II.7 se muestra cómo se representan los elementos almacenados en las listas (correspondientes al nombre de las clases identificadas y su descripción en lenguaje natural) en el formato *X- WebODE*.

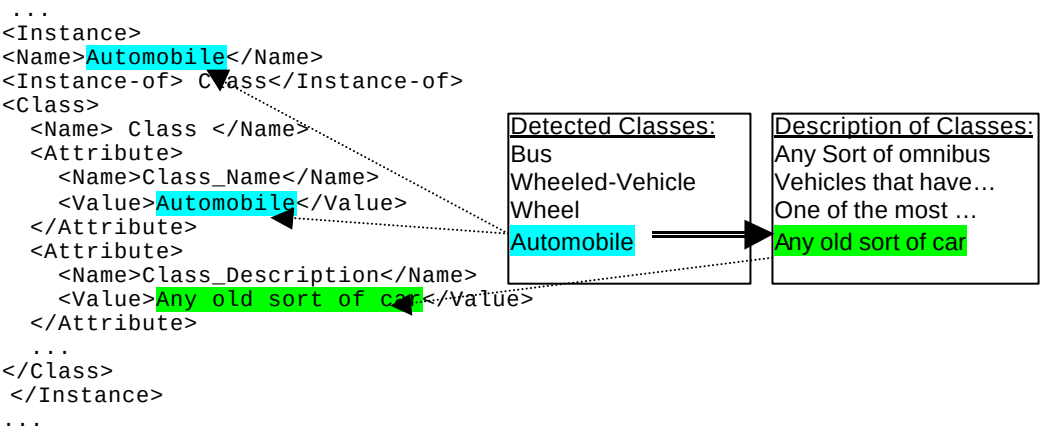


Figura II.7. Creación del fichero en *X-WebODE* a partir de la información recogida por los *parsers* del *Ontolingua_Ontowrapper*.

II.3. Extracción del Contenido desde Ontologías Implementadas en *DAML+OIL*

Se ha desarrollado una aplicación software, denominada ***DAML_OntoWrapper***, que extrae la información relevante de ontologías implementadas en el lenguaje *DAML+OIL*, y genera ficheros *XML* en formato *X-WebODE*, que son compatibles con el modelo conceptual de la dimensión "contenido" de la RO. Al igual que en la sección anterior, se muestra en este apartado una breve descripción de la sintaxis de *DAML+OIL*, un ejemplo de los patrones sintácticos utilizados y un ejemplo muy simple de su funcionamiento.

II.3.1. El lenguaje *DAML+OIL*

DAML+OIL [Hor01] es un lenguaje para la *web* semántica que está siendo desarrollado conjuntamente por Europa y Estados Unidos. Es un lenguaje basado en *XML* que permite representar conceptos, taxonomías, relaciones binarias, e instancias, y se está realizando un gran esfuerzo por incorporar mecanismos de razonamiento. Estos términos van a ser detectados por la aplicación *DAML_OntoWrapper*, y los va a transformar en instancias válidas del modelo conceptual de la RO en la dimensión "contenido". En este apartado se muestran formatos de instrucciones básicas de *DAML+OIL* para aclarar los siguientes ejemplos de extracción de información, pero la sintaxis completa se puede encontrar en la URL: http://www.daml.org/2001/03/reference.html.

Para poder seguir el posterior ejemplo de la figura II.9, en la figura II.8 se muestran los formatos en los que pueden representarse las **clases** en *DAML+OIL*:

Formato 1:

```
<daml:Class rdf:ID="nombre_de_la_clase">
  [<rdfs:label>descripción de clase </rdfs:label>]
  [<rdfs:comment>descripción de clase </rdfs:comment>]
  [<rdfs:subClassOf rdf:resource="#superclase" />]*
  [<daml:disjointFrom rdfs:resource="#superclase_disjunta"/>]*
  [<daml:disjointUnionOf rdfs:resource="#superclase_exhaustiva"/>]*
</daml:Class>
```

Formato 2:

```
<daml:Class rdf:about="&rdfs;nombre_de_la_clase">
  [<rdfs:label>descripción de clase </rdfs:label>]
  [<rdfs:comment>descripción de clase</rdfs:comment>]
  [<rdfs:subClassOf rdf:resource="&rdfs;superclase" />]*
  [<daml:disjointFrom rdf:resource="&rdfs;superclase_disjunta"/>]*
  [<daml:disjointUnionOf resource="&rdfs;superclase_exhaustiva"/>]*
</daml:Class>
```

Figura II.8. Formatos de definición en *DAML+OIL* de una clase.

Los términos `label` y `comment` hacen referencia a la descripción en lenguaje natural de la clase, pudiéndose utilizar ambos indistintamente. Para indicar las superclases de la clase que se está definiendo se utiliza el término `subClassOf` y para definir clases disjuntas y exhaustivas se

utilizan las palabras clave `disjointFrom` y `disjointUnionOf`, respectivamente. En la cabecera del documento debe estar descrito el espacio de nombres [Las99] de *rdf*, *rdfs* y *daml* de la siguiente forma:

```
xmlns: rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns: rdfs: http://www.w3.org/2000/01/rdf-schema#
daml: http://www.daml.org/2001/03/daml+oil#
```

Figura II.9. Definición de los espacios de nombres para documentos *DAML+OIL* .

Los siguientes ejemplos de la figura II.10 utilizan sendos formatos de instrucciones para definir dos clases, que aparecen en las ontologías *Publication* y *SWRC Ontology* respectivamente, accesibles desde la *DAML Ontology Library*:

```
<daml:Class rdf:ID="Author"
       <rdf:comment>Author of a publication</rdf:comment>
       <rdf:subClassOf rdf:resource="#Person"/>
</daml:Class>

<daml:Class rdf:about="&c;Conference"
       <rdf:comment>Describes a generic concept about events </rdf:comment>
       <rdf:subClassOf rdf:resource="&c;Event"/>
       <daml:disjointFrom rdf:resource="&c;Workshop"/>
</daml:Class>
```

Figura II.10. Dos ejemplos de representación de clases en *DAML+OIL* .

II.3.2. Patrones sintácticos para la extracción de información desde *DAML+OIL*

Se han identificado los patrones sintácticos a partir de la sintaxis del lenguaje *DAML+OIL*, y se ha implementado el *DAML_OntoWrapper*. Como ejemplo, en las figuras II.11.a y II.11.b se muestran los patrones sintácticos para detectar la especificación de una clase, que completa la característica *Class* en la RO; en la figura II.11.c y II.11.d se muestra cómo se detecta la descripción en lenguaje natural de un concepto, que completa la característica *Class_Description* en la RO. Se ha seguido la notación de autómatas finitos para expresar estos patrones sintácticos.

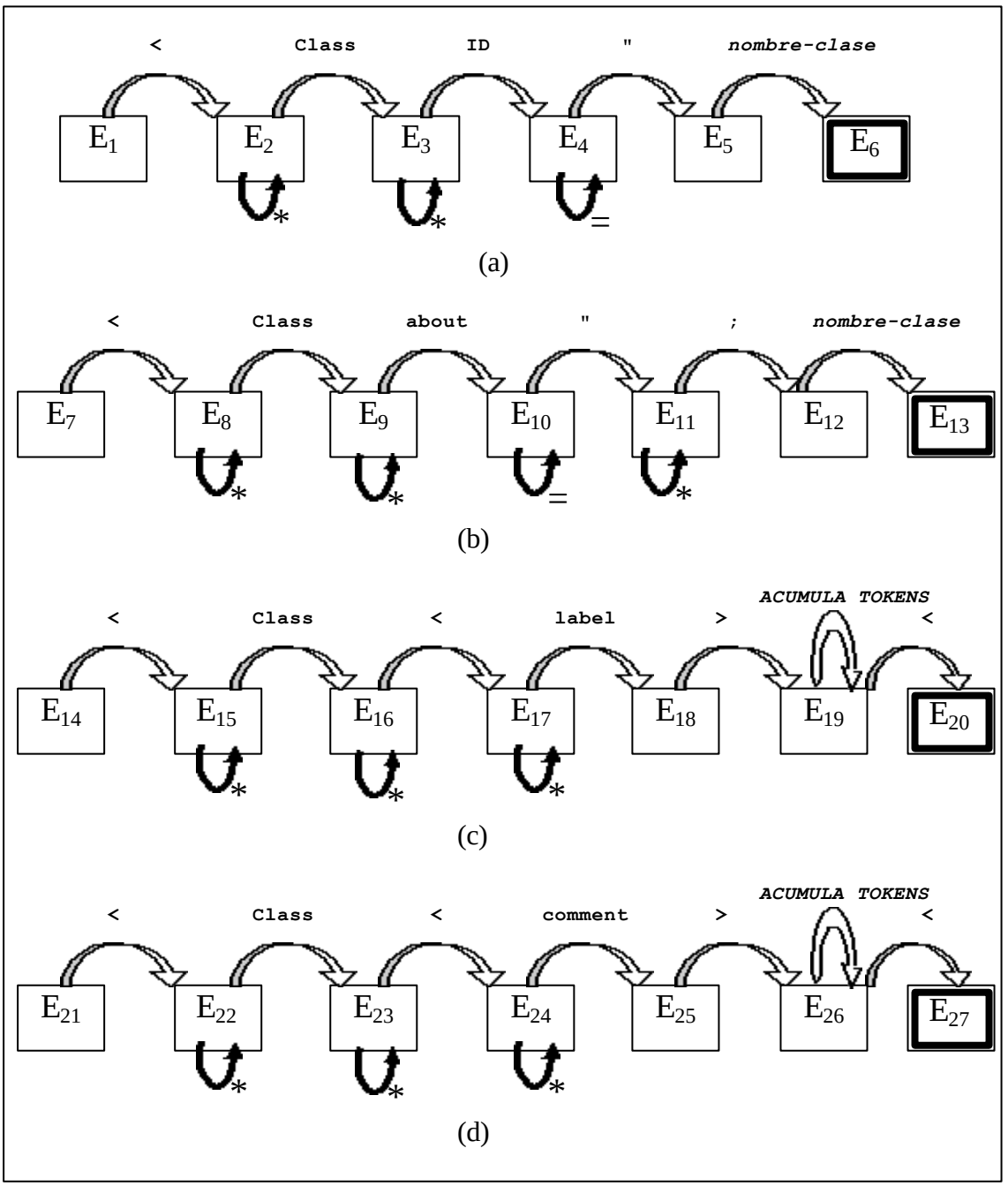


Figura II.11. Patrones sintácticos para detectar: (a) y (b) definición de clases; (c) descripción de una clase con la etiqueta `label`; y (d) descripción de una clase con la etiqueta `comment`.

II.3.3. Funcionamiento de la aplicación *DAML_OntoWrapper* y ejemplo de detección de tokens

DAML_OntoWrapper implementa los patrones sintácticos para *DAML+OIL*. Se muestra, como ejemplo muy básico de detección de tokens, la detección de la clase *Conference* incluida en la ontología *SWRC Ontology*. Se usarán los patrones sintácticos que se definen en la figura

II.11.

Detección de clases: Usando los patrones de transformación de la figura II.11.a y figura II.11.b, el analizador sintáctico recorre el fichero de la ontología implementada en *DAML+OIL* buscando términos de tipo "clase". En una estructura de lista se almacenan los *tokens* de este tipo encontrados hasta ese momento.

En el ejemplo de la figura II.12, utilizando el patrón de transformación de la figura II.11.b, se detecta el nombre de la clase y se almacena en la lista de conceptos "clase" encontrados.

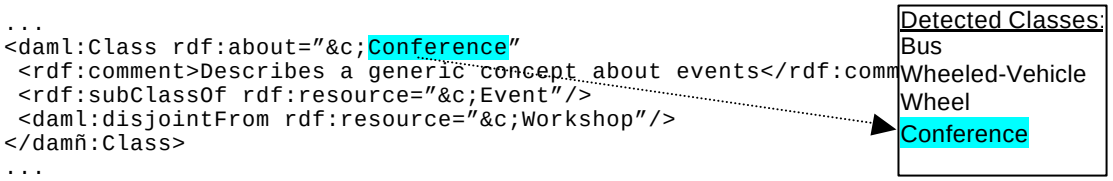


Figura II.12. Inserción, en la estructura de tipo lista, de la clase detectada por el *DAML_Ontowrapper*.

Detección de la descripción de una clase: usando en este caso el patrón de transformación de la figura II.11.d, el parser correspondiente detecta la descripción en lenguaje natural de la clase identificada y la almacenada en la correspondiente lista de descripción de clases, como se muestra en la figura II.13.

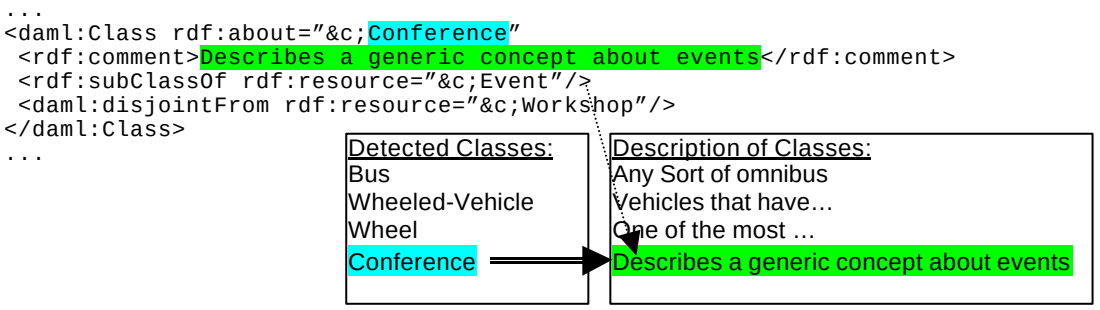


Figura II.13. Inserción, en la estructura de tipo lista, de la descripción en lenguaje natural de la clase detectada por el *DAML_Ontowrapper*.

El proceso de generación del fichero *X-WebODE* es idéntico al implementado en el *Ontolingua-OntoWrapper*, que se muestra en al apartado II.2.4.

II.4. Extracción del Contenido desde Ontologías Implementadas en *X-WebODE*

Las ontologías construidas con el entorno de desarrollo *WebODE* pueden ser exportadas al lenguaje de intercambio de datos *X-WebODE*. Se puede extraer la información relativa al "contenido" de la RO desde estos ficheros utilizando la aplicación ***X-WebODE_OntoWrapper***. Este *OntoWrapper* trata el mismo formato de representación en el fichero de entrada y en el de salida, ya que tiene como entrada un fichero en *X-WebODE* con una ontología, y como salida una instancia del modelo conceptual de la RO de la dimensión "contenido", que se representará también en *X-WebODE*, para que pueda ser importada por el entorno *WebODE*. Esta aplicación tiene un funcionamiento semejante a los *OntoWrappers* descritos en los dos apartados anteriores, con la única salvedad de que las ontologías que se analizan están representadas en el propio lenguaje *X-WebODE*. Se muestra a continuación un ejemplo de los patrones sintácticos identificados, y un ejemplo muy simple de funcionamiento del *X-WebODE_OntoWrapper*.

II.4.1. El formato de representación *X-WebODE*

La sintaxis básica de *X-WebODE* se describió en el apartado II.1., y también se indicó dónde puede encontrarse la descripción completa. En la figura II.14 se muestra el formato abreviado que sigue la *DTD* para representar una **clase**.

```
<Concept>
   <Name>nombre_de_la_clase</Name>
   [<Description>descripción_de_la_clase</Description>]+
   [<Class-Attribute>definición_de_atr_de_clase</Class-Attribute>]*
   [<Instance-Attribute>definición_atr_de_instancia</Instance-Attribute>]*
   [<Synonym>definición_de_sinónimo</Synonym>]*
   [<Abbreviation>definición_de_abreviación</Abbreviation>]*
   [<Related-Reference>definición_de_referencia</Related-Reference>]*
   [<Related-Formula>definición_de_fórmula</Related-Formula>]*
</Concept>
```

Figura II.14. Formatos de definición en *X-WebODE* del concepto "clase".

El término `Description` hace referencia a la descripción en lenguaje natural de la clase. Dentro de las etiquetas `Class-Attribute` e `Instance-Attribute` se definen sus atributos de clase y de instancia. Los sinónimos, las abreviaturas y las referencias de la clase dentro de las etiquetas `Synonym`, `Abbreviation` y `Related-Reference`, respectivamente; y las fórmulas relacionadas en `Related-Formula`.

El siguiente ejemplo de la figura II.15 muestra la clase *Business trip* que se encuentra definida en la ontología *Springer Book Ontology*, almacenada en el entorno *WebODE*:

```
<Concept>
      <Name>Business trip</Name>
      <Description>A special trip for businessmen</Description>
      <Instance-Attribute>
            <Name>budget</Name>
            <Type>Float</Type>
            <Minimum-Cardinality>0</Minimum-Cardinality>
            <Maximum-Cardinality>1</Maximum-Cardinality>
            <Minimum-Value>1000</Minimum-Value>
            <Maximum-Value>3000</Maximum-Value>
      </Instance-Attribute>
</Concept>
```

Figura II.15. Ejemplo de definición de una clase en *X-WebODE*.

II.4.2. Patrones sintácticos para la extracción de información desde *X-WebODE*

Se han identificado los patrones sintácticos a partir de la *DTD* de *X-WebODE*, y se ha implementado el correspondiente *OntoWrapper*, denominado *X-WebODE_OntoWrapper* que sigue la especificación de estos patrones. Como ejemplo, en la figura II.16.a se muestra el patrón de transformación para detectar la especificación de una clase; en la figura II.16.b se muestra cómo se detecta la descripción en lenguaje natural de una clase. Se ha seguido la notación de autómatas finitos para expresar estos patrones sintácticos.

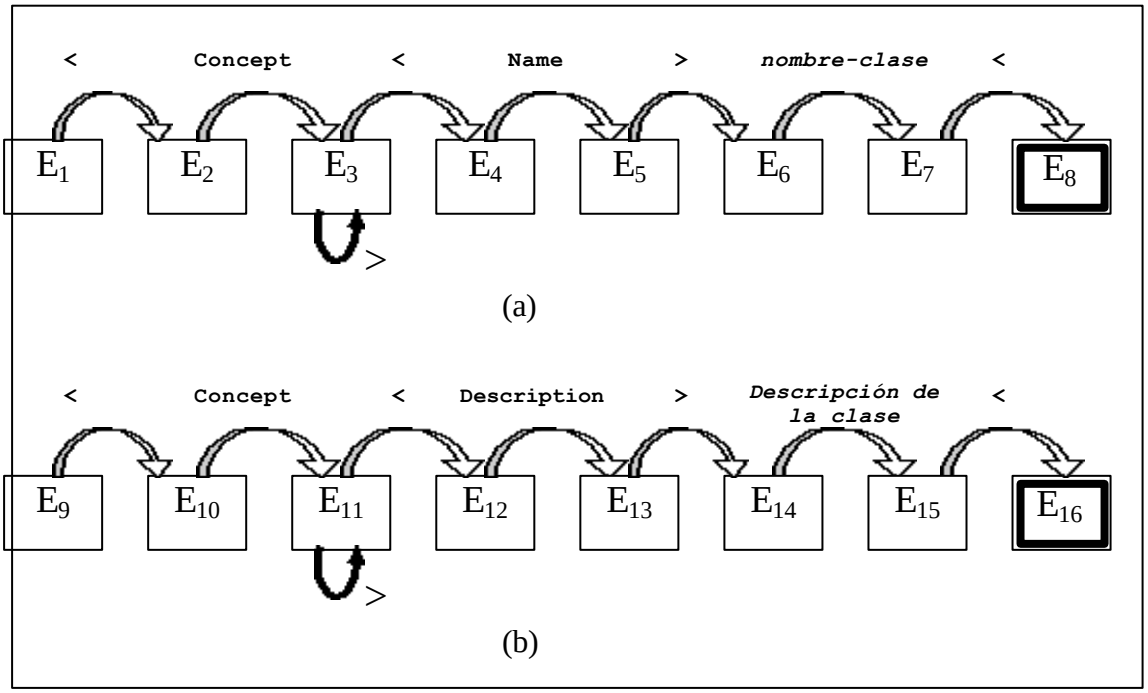


Figura II.16. Patrones sintácticos para detectar: (a) definición de clases; y (b) descripción de una clase.

II.4.3. Funcionamiento de la aplicación *X-WebODE_OntoWrapper* y ejemplo de detección de tokens

De forma similar al funcionamiento de *Ontolingua_OntoWrapper* y *DAML_OntoWrapper, X-WebODE_OntoWrapper* implementa los patrones sintácticos para *X-WebODE*. Se muestra, como ejemplo muy básico de detección de *tokens*, la detección de la clase *Business trip* incluida en la ontología *Springer Book Ontology*. Se usan los patrones que se definen en la figura II.16.

Detección de clases: Usando el patrón de transformación de la figura II.16.a, el analizador sintáctico recorre el fichero de la ontología implementada en *X-WebODE* buscando términos de tipo "clase". Como se muestra en la figura II.17, en una estructura de lista se almacenan los tokens de este tipo encontrados hasta ese momento; utilizando el patrón de la figura II.16.a, se detecta el nombre de la clase y se almacena en la lista de conceptos "clase" encontrados.

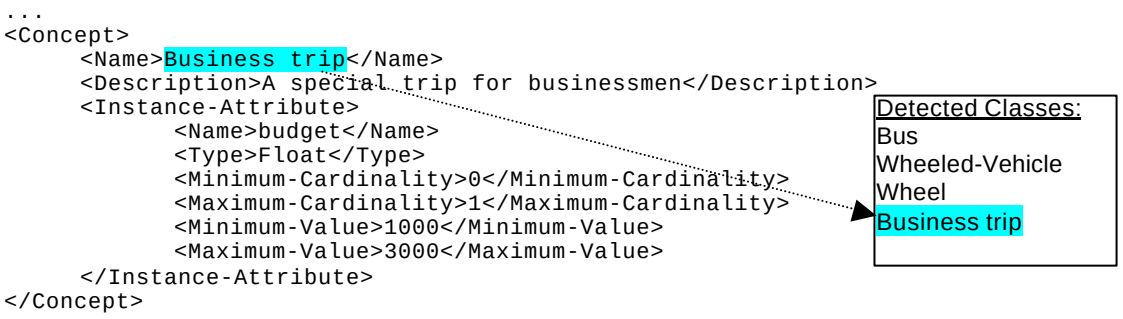


Figura II.17. Inserción, en la estructura de tipo lista, de la clase detectada por el *X-WebODE_Ontowrapper*.

Detección de la descripción de una clase: usando en este caso el patrón de transformación de la figura II.16.b, el *parser* correspondiente detecta (figura II.18) la descripción en lenguaje natural de la clase y la almacenada en la correspondiente lista de descripción de clases.

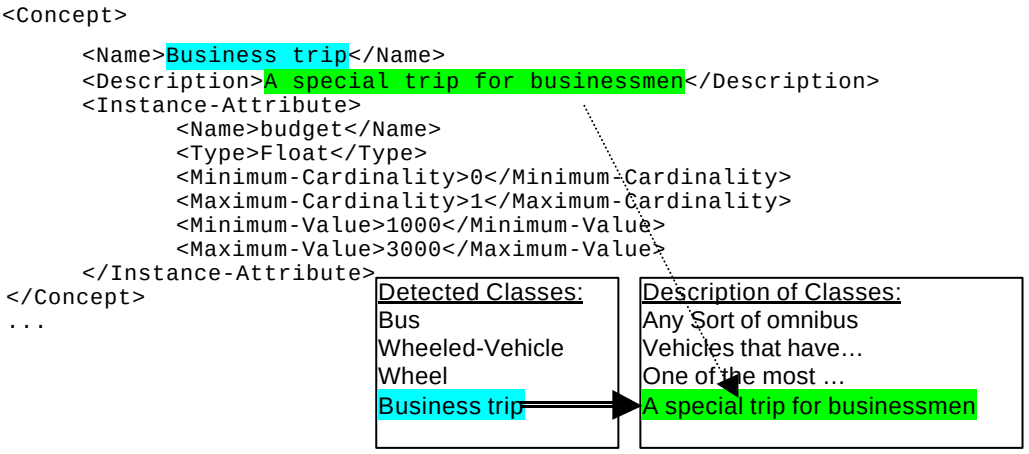


Figura II.18. Inserción, en la estructura de tipo lista, de la descripción en lenguaje natural de la clase detectada por el *X-WebODE_Ontowrapper*.

El proceso de generación del fichero *X-WebODE* es idéntico al implementado en el *Ontolingua-OntoWrapper*, que se muestra en al apartado II.2.4.

Anexo III

Cálculo de los Pesos por Defecto a partir de los Cuestionarios de Opinión

A continuación se muestran los resultados de las encuestas de opinión, sobre características de ontologías, realizadas a 10 usuarios familiarizados con el desarrollo de ontologías. Las preguntas son las expuestas en el cuestionario del anexo I de este trabajo.

La tabla III.I muestra, en la primera columna, el código de la característica, factor o dimensión del que se está preguntando el grado de importancia. Este código coincide con el indicado en las tablas 4.1, 4.3, 4.4, 4.5, y 4.6 que identifican los criterios del marco multinivel de características. En la columna segunda se indican las dimensiones, factores y principales características a modo de guía. Desde las columnas tercera a decimotercera se representan los valores de importancia que los entrevistados han indicado en los 160 criterios. Toman los valores: *no es importante (1)*, *no es fundamental (2)*, *es importante (3)*, *es muy importante (4)*, *es fundamental (5)*, o *no contesta (NC)*. La columna decimoquinta muestra el valor medio de los valores de importancia asignados por los usuarios que han completado la pregunta. La columna decimosexta muestra la desviación típica de los resultados, y la columna decimoctava el error medio. La columna vigésima muestra los pesos medios asignados al criterio, normalizados con los criterios hermanos (aquellos que tienen el mismo criterio padre en el árbol multinivel de características). Estos últimos serán los pesos por defecto que proporciona el método OntoMetric.

El valor medio de la desviación típica de los valores de cada variable es de 0.9. Puede considerarse un valor bajo para la desviación típica, ya que los valores de las distintas variables se alejan poco del valor de la media, aunque hay que tener en cuenta que estamos trabajando únicamente con diez muestras y con sólo cinco posibles valores para cada variable.

La media obtenida de los valores del error medio es de 0.32. Puede afirmarse que no hay muchas variaciones en las respuestas de los encuestados. Estas diferencias pueden explicarse por las diversas percepciones de los encuestados acerca de los diferentes aspectos que los usuarios buscan a la hora de usar una ontología.

		Socorro Bernardos	Mercedes Blázquez	Oscar Corcho	Raúl de Diego	Juan Manuel Dodero	Mariano Fernández	Asunción Gómez	Vanesa López	Adolfo Lozano	Jose Angel Ramos	**Pesos medios**	Desviación típica	Error medio	**Valores normalizados**
C	**CONTENIDO**	**3**	**4**	**4**	**4**	**3**	**3**	**3**	**5**	**5**	**3**	**3,7**	**0,8**	**0,26**	**0,2241**
C1	**Conceptos/ Instancias / Hechos/ Afirmaciones**	**3**	**3**	**3**	**2**	**4**	**3**	**3**	**3**	**4**	**3**	**3,1**	**0,6**	**0,18**	**0,2164**
C11		2	3	3	4	2	3	4	4	4	5	**3,4**	1	0,31	0,1694
C12		2	NC	4	3	2	2	3	NC	2	2	**2,5**	0,8	0,27	0,1246
C13		5	3	3	5	4	5	3	NC	2	NC	**3,75**	1,2	0,41	0,1869
C14		5	3	5	5	NC	5	3	NC	4	NC	**4,29**	1	0,36	0,2135
C15		4	4	4	4	2	3	5	5	4	3	**3,8**	0,9	0,29	0,1893
C16		1	2	4	3	1	3	2	NC	1	4	**2,33**	1,2	0,41	0,1163
C2	**Relaciones**	**3**	**NC**	**3**	**5**	**2**	**5**	**4**	**5**	**3**	**4**	**3,78**	**1,1**	**0,36**	**0,2638**
C21		2	3	3	5	3	3	4	4	3	4	**3,4**	0,8	0,27	0,1465
C22		3	3	3	2	2	3	4	3	2	4	**2,9**	0,7	0,23	0,1250
C23		5	5	4	5	NC	5	4	NC	3	3	**4,25**	0,9	0,31	0,1831
C24		NC	4	5	4	1	5	5	NC	2	NC	**3,71**	1,6	0,61	0,1600
C25		NC	NC	2	3	3	3	3	1	1	4	**2,5**	1,1	0,38	0,1077
C26		1	3	4	3	3	3	2	NC	1	1	**2,33**	1,1	0,37	0,1005
C3	**Taxonomía**	**NC**	**3**	**4**	**5**	**4**	**5**	**4**	**4**	**4**	**4**	**4,11**	**0,6**	**0,2**	**0,2870**
C31		4	4	4	3	NC	3	4	NC	2	2	**3,25**	0,9	0,31	0,2302
C32		1	NC	1	1	2	2	2	3	1	2	**1,67**	0,7	0,24	0,1181
C33		2	3	3	4	4	3	3	1	1	4	**2,8**	1,1	0,36	0,1983
C34		2	3	3	4	4	3	4	1	1	4	**2,9**	1,2	0,38	0,2054
C35		1	NC	1	4	3	1	1	1	1	NC	**1,63**	1,2	0,42	0,1151
C36		1	NC	2	3	3	1	2	1	2	NC	**1,88**	0,8	0,3	0,1328
C4	**Axiomas**	**3**	**4**	**3**	**3**	**3**	**3**	**4**	**NC**	**3**	**4**	**3,33**	**0,5**	**0,17**	**0,2327**
C41		NC	3	4	1	3	3	2	NC	2	4	**2,75**	1	0,37	0,1959
C42		NC	4	3	3	3	3	4	NC	2	NC	**3,14**	0,7	0,26	0,2239
C43		NC	4	5	4	NC	3	4	NC	3	5	**4**	0,8	0,31	0,2850
C44		NC	3	2	1	1	2	4	NC	2	NC	**2,14**	1,1	0,4	0,1527
C45		1	4	2	1	NC	3	3	NC	1	1	**2**	1,2	0,42	0,1425
M	**METODOLOGÍAS**	**2**	**3**	**3**	**1**	**2**	**5**	**4**	**NC**	**3**	**1**	**2,67**	**1,3**	**0,44**	**0,1615**
M1	**Precisión**	**4**	**4**	**2**	**1**	**2**	**5**	**4**	**3**	**2**	**3**	**3**	**1,2**	**0,39**	**0,3089**
M11		4	4	2	1	1	5	3	NC	3	3	**2,89**	1,4	0,45	0,1871
M12		4	4	2	1	3	5	3	NC	3	3	**3,11**	1,2	0,39	0,2014
M13		4	4	3	1	3	4	3	NC	2	NC	**3**	1,1	0,38	0,1942
M14		4	4	2	1	3	4	4	NC	3	4	**3,22**	1,1	0,36	0,2086
M15		4	4	3	1	2	5	4	3	3	NC	**3,22**	1,2	0,4	0,2086
M2	**Usabilidad**	**4**	**4**	**2**	**3**	**3**	**4**	**4**	**5**	**3**	**4**	**3,6**	**0,8**	**0,27**	**0,3707**
M21		4	5	2	1	4	4	4	5	4	5	**3,8**	1,3	0,42	0,3311
M22		4	5	3	3	3	4	4	5	4	4	**3,9**	0,7	0,23	0,3398
M23		3	4	5	3	3	3	5	NC	3	5	**3,78**	1	0,32	0,3291
M3	**Madurez**	**3**	**4**	**3**	**3**	**2**	**5**	**4**	**NC**	**2**	**2**	**3,11**	**1,1**	**0,35**	**0,3204**
M31		3	3	4	3	2	3	5	NC	2	2	**3**	1	0,33	0,2967
M32		3	4	4	3	4	4	5	NC	2	2	**3,44**	1	0,34	0,3407
M33		3	4	4	3	3	5	5	NC	3	3	**3,67**	0,9	0,29	0,3626

Tabla III.1. Resultados de la encuesta de opinión sobre criterios de ontologías **(1/4)**.

		Socorro Bernardos	Mercedes Blázquez	Oscar Corcho	Raúl de Diego	Juan Manuel Dodero	Mariano Fernández	Asunción Gómez	Vanesa López	Adolfo Lozano	Jose Angel Ramos		**Pesos medios**	Desviación típica	Error medio	**Valores normalizados**
T	**ENTORNOS**	**3**	**4**	**5**	**3**	**5**	**4**	**4**	**4**	**5**	**4**		**4,1**	**0,7**	**0,23**	**0,2484**
T1	**Prestaciones**	**4**	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**4**		**4,1**	**0,7**	**0,23**	**0,1316**
T11		4	3	2	3	1	3	3	NC	2	3		**2,67**	0,9	0,29	0,1316
T12		3	3	2	3	1	3	4	3	2	2		**2,6**	0,8	0,27	0,1283
T13		3	3	2	3	1	3	4	4	2	2		**2,7**	0,9	0,3	0,1332
T14		5	5	4	5	3	5	3	5	3	5		**4,3**	0,9	0,3	0,2122
T15		4	5	4	5	3	5	3	4	2	3		**3,8**	1	0,33	0,1875
T16		5	5	4	4	4	5	5	3	3	4		**4,2**	0,8	0,25	0,2072
T2	**Visualización**	**4**	**4**	**4**	**5**	**4**	**5**	**4**	**3**	**3**	**5**		**4,1**	**0,7**	**0,23**	**0,1316**
T21		4	4	5	5	4	5	4	4	5	4		**4,4**	0,5	0,16	0,2651
T22		4	5	4	4	3	5	4	3	4	4		**4**	0,7	0,21	0,2410
T23		4	2	5	5	4	5	5	5	3	4		**4,2**	1	0,33	0,2534
T24		4	5	4	5	4	5	5	2	2	4		**4**	1,2	0,37	0,2415
T3	**Edición**	**4**	**4**	**5**	**5**	**3**	**4**	**4**	**4**	**5**	**3**		**4,1**	**0,7**	**0,23**	**0,1316**
T31		4	5	5	5	4	3	5	4	4	NC		**4,33**	0,7	0,24	0,2723
T32		4	4	5	5	3	3	5	5	5	3		**4,2**	0,9	0,29	0,2640
T33		3	4	4	5	4	3	5	NC	3	3		**3,78**	0,8	0,28	0,2374
T34		3	3	4	5	4	4	5	2	3	3		**3,6**	1	0,31	0,2263
T4	**Interacción**	**5**	**3**	**4**	**5**	**2**	**3**	**5**	**3**	**3**	**NC**		**3,67**	**1,1**	**0,37**	**0,1177**
T41		5	3	3	3	2	3	5	5	3	2		**3,4**	1,2	0,37	0,1901
T42		5	4	4	4	3	3	4	4	2	4		**3,7**	0,8	0,26	0,2068
T43		4	4	3	5	3	3	4	4	4	5		**3,9**	0,7	0,23	0,2180
T44		3	2	3	5	4	2	4	NC	3	3		**3,22**	1	0,32	0,1801
T45		3	3	4	5	4	3	4	NC	4	3		**3,67**	0,7	0,24	0,2056
T5	**Aspectos metodológicos**	**5**	**3**	**3**	**1**	**4**	**4**	**5**	**NC**	**2**	**3**		**3,33**	**1,3**	**0,44**	**0,1071**
T51		3	3	3	1	2	1	3	NC	2	4		**2,44**	1	0,34	0,1832
T52		4	3	5	1	3	4	4	3	3	5		**3,5**	1,2	0,37	0,2623
T53		4	4	4	1	4	3	5	3	3	3		**3,4**	1,1	0,34	0,2548
T54		4	3	5	3	4	3	4	5	4	5		**4**	0,8	0,26	0,2998
T6	**Aspectos cooperativos**	**2**	**3**	**5**	**4**	**4**	**3**	**5**	**4**	**4**	**2**		**3,6**	**1,1**	**0,34**	**0,1156**
T61		2	3	5	4	3	3	5	4	3	3		**3,5**	1	0,31	0,1711
T62		2	2	4	4	3	3	3	4	2	NC		**3**	0,9	0,29	0,1467
T63		2	4	3	4	5	3	2	1	3	2		**2,9**	1,2	0,38	0,1418
T64		2	4	5	4	5	3	5	NC	1	5		**3,78**	1,5	0,49	0,1847
T65		2	4	5	4	4	2	4	3	3	4		**3,5**	1	0,31	0,1711
T66		4	4	5	4	3	4	5	NC	2	3		**3,78**	1	0,32	0,1847
T7	**Traducción**	**4**	**3**	**5**	**5**	**4**	**5**	**5**	**NC**	**3**	**NC**		**4,25**	**0,9**	**0,31**	**0,1364**
T71		NC	4	4	5	4	3	4	4	4	NC		**4**	0,5	0,19	0,1733
T72		NC	4	4	5	5	4	4	4	4	NC		**4,25**	0,5	0,16	0,1841
T73		4	4	5	5	3	3	4	1	2	NC		**3,44**	1,3	0,44	0,1492
T74		3	4	4	5	5	4	5	4	2	NC		**4**	1	0,33	0,1733
T75		4	4	5	5	4	4	5	NC	3	NC		**4,25**	0,7	0,25	0,1841
T76		4	NC	3	4	3	3	3	NC	2	NC		**3,14**	0,7	0,26	0,1361
T8	**Integración**	**4**	**3**	**5**	**5**	**4**	**4**	**4**	**NC**	**3**	**NC**		**4**	**0,8**	**0,27**	**0,1284**
T81		3	3	4	5	3	4	4	NC	2	NC		**3,5**	0,9	0,33	0,1595
T82		3	4	4	5	3	4	5	NC	1	NC		**3,63**	1,3	0,46	0,1652
T83		NC	4	5	4	4	4	4	5	2	NC		**4**	0,9	0,33	0,1823
T84		4	3	5	5	4	5	4	4	4	NC		**4,22**	0,7	0,22	0,1924
T85		3	2	4	5	3	4	4	NC	2	NC		**3,38**	1,1	0,38	0,1538
T86		3	2	4	5	3	4	3	3	2	NC		**3,22**	1	0,32	0,1468

Tabla III.1. Resultados de la encuesta de opinión sobre criterios de ontologías **(2/4)**.

		Socorro Bernardos	Mercedes Blázquez	Oscar Corcho	Raúl de Diego	Juan Manuel Dodero	Mariano Fernández	Asunción Gómez	Vanesa López	Adolfo Lozano	Jose Angel Ramos		Pesos medios	Desviación típica	Error medio	Valores normalizados
L	**LENGUAJES**	**3**	**4**	**5**	**2**	**3**	**5**	**5**	**2**	**4**	NC		**3,67**	**1,2**	**0,41**	**0,2221**
L1	**REPRES. DEL CONOCIMIENTO**	**5**	**3**	**3**	**5**	**3**	**3**	**5**	**NC**	**3**	**NC**		**3,75**	**1**	**0,37**	**0,5455**
L11	**Conceptos**	**5**	**4**	**5**	**5**	**5**	**5**	**5**	**NC**	**4**	**NC**		**4,75**	**0,5**	**0,16**	**0,1658**
L111		5	5	5	5	5	5	5	5	5	NC		**5**	0	0	0,3172
L112		NC	4	4	3	4	4	5	NC	3	NC		**3,86**	0,7	0,26	0,2447
L113		NC	2	4	1	**2**	2	5	NC	2	NC		**2,57**	1,4	0,53	0,1631
L114		NC	NC	4	1	NC	3	4	NC	3	NC		**3**	1,2	0,55	0,1903
L115		NC	2	1	1	NC	1	2	NC	1	NC		**1,33**	0,5	0,21	0,0846
L12	**Atributos**	**4**	**4**	**4**	**5**	**4**	**5**	**5**	**3**	**5**	**NC**		**4,33**	**0,7**	**0,24**	**0,1513**
L121		4	3	4	5	4	3	5	5	3	NC		**4**	0,9	0,29	0,1906
L122		5	3	5	5	4	5	5	5	5	NC		**4,67**	0,7	0,24	0,2223
L123		NC	3	4	*1*	3	3	5	1	5	NC		**3,13**	1,6	0,55	0,1489
L124		NC	NC	3	*1*	4	2	4	2	2	NC		**2,57**	1,1	0,43	0,1225
L125		4	NC	3	4	3	4	4	1	3	NC		**3,25**	1	0,37	0,1548
L126		4	3	3	4	3	4	4	NC	2	NC		**3,38**	0,7	0,26	0,1608
L13	**Facetas**	**NC**	**NC**	**5**	**5**	**3**	**5**	**4**	**NC**	**4**	**NC**		**4,33**	**0,8**	**0,33**	**0,1513**
L131		NC	4	3	4	4	3	4	3	3	NC		**3,5**	0,5	0,19	0,1867
L132		NC	5	5	5	3	5	5	3	4	NC		**4,38**	0,9	0,32	0,2334
L133		NC	4	5	5	3	4	5	3	5	NC		**4,25**	0,9	0,31	0,2267
L134		NC	NC	4	4	3	2	4	NC	3	NC		**3,33**	0,8	0,33	0,1778
L135		NC	NC	3	4	3	3	3	2	5	NC		**3,29**	1	0,36	0,1753
L14	**Relaciones**	**4**	**3**	**5**	**5**	**4**	**2**	**5**	**3**	**5**	**NC**		**4**	**1,1**	**0,37**	**0,1397**
L141		4	2	5	1	3	3	5	1	4	NC		**3,11**	1,5	0,51	0,1418
L142		3	2	3	4	4	2	3	3	2	NC		**2,89**	0,8	0,26	0,1317
L143		4	4	5	5	4	2	5	5	5	NC		**4,33**	1	0,33	0,1975
L144		3	2	3	3	NC	3	3	3	2	NC		**2,75**	0,5	0,16	0,1253
L145		NC	4	5	3	3	5	3	NC	3	NC		**3,71**	1	0,36	0,1693
L146		NC	3	4	3	3	4	3	NC	2	NC		**3,14**	0,7	0,26	0,1432
L147		NC	NC	1	1	NC	2	3	NC	3	NC		**2**	1	0,45	0,0912
L15	**Taxonomía**	**4**	**3**	**5**	**5**	**4**	**5**	**5**	**4**	**4**	**NC**		**4,33**	**0,7**	**0,24**	**0,1513**
L151		4	3	5	5	3	5	5	5	5	NC		**4,44**	0,9	0,29	0,2079
L152		NC	3	2	3	2	2	2	5	2	NC		**2,63**	1,1	0,38	0,1228
L153		4	3	4	4	3	3	4	2	3	NC		**3,33**	0,7	0,24	0,1559
L154		3	3	4	4	3	3	4	2	3	NC		**3,22**	0,7	0,22	0,1507
L155		NC	3	4	5	3	5	5	3	3	NC		**3,88**	1	0,35	0,1813
L156		NC	3	4	5	3	5	5	3	3	NC		**3,88**	1	0,35	0,1813
L16	**Axiomas**	**4**	**3**	**4**	**4**	**3**	**4**	**5**	**NC**	**3**	**NC**		**3,75**	**0,7**	**0,25**	**0,1309**
L161		NC	3	3	1	2	3	4	NC	4	NC		**2,86**	1,1	0,4	0,2925
L162		NC	2	1	1	2	2	2	NC	1	NC		**1,57**	0,5	0,2	0,1609
L163		NC	2	4	1	2	3	4	NC	3	NC		**2,71**	1,1	0,42	0,2779
L164		2	2	3	3	2	3	3	NC	3	NC		**2,63**	0,5	0,18	0,2687
L17	**Reglas de producción**	**NC**	**2**	**3**	**4**	**2**	**4**	**4**	**NC**	**3**	**NC**		**3,14**	**0,9**	**0,34**	**0,1097**
L171		NC	4	2	1	NC	2	3	NC	3	NC		**2,5**	1	0,43	0,1716
L172		NC	2	1	1	NC	2	2	NC	2	NC		**1,67**	0,5	0,21	0,1144
L173		NC	2	2	3	NC	3	2	NC	2	NC		**2,33**	0,5	0,21	0,1602
L174		NC	2	3	3	NC	3	3	NC	2	NC		**2,67**	0,5	0,21	0,1831
L175		NC	NC	2	3	NC	3	3	NC	2	NC		**2,6**	0,5	0,24	0,1785
L176		NC	NC	3	3	NC	3	2	NC	3	NC		**2,8**	0,4	0,2	0,1922

Tabla III.1. Resultados de la encuesta de opinión sobre criterios de ontologías **(3/4)**.

		Socorro Bernardos	Mercedes Blázquez	Oscar Corcho	Raúl de Diego	Juan Manuel Dodero	Mariano Fernández	Asunción Gómez	Vanesa López	Adolfo Lozano	Jose Angel Ramos		**Pesos medios**	Desviación típica	Error medio		**Valores normalizados**
L2	**MECANISMO DE INFERENCIA**	**3**	**3**	**3**	**3**	**3**	**5**	**2**	**NC**	**3**	**NC**		**3,13**	**0,8**	**0,3**		**0,4545**
L21	**Potencial de razonamiento**	**3**	**3**	**4**	**4**	**2**	**5**	**3**	**NC**	**4**	**NC**		**3,5**	**0,9**	**0,33**		**0,5158**
L211		3	3	4	5	2	5	3	3	3	NC		**3,44**	1	0,34		0,1811
L212		3	1	3	1	2	5	2	NC	3	NC		**2,5**	1,3	0,46		0,1315
L213		3	2	3	1	2	4	2	NC	1	NC		**2,25**	1	0,37		0,1183
L214		3	NC	3	4	2	4	3	NC	3	NC		**3,14**	0,7	0,26		0,1653
L215		3	3	3	4	2	4	4	NC	3	NC		**3,25**	0,7	0,25		0,1709
L216		3	NC	1	3	3	4	3	NC	2	NC		**2,71**	1	0,36		0,1427
L217		NC	1	2	1	2	2	2	NC	2	NC		**1,71**	0,5	0,18		0,0902
L22	**Motor de inferencia**	**NC**	**3**	**4**	**3**	**2**	**4**	**4**	**NC**	**3**	**NC**		**3,29**	**0,8**	**0,29**		**0,4842**
L221		NC	4	4	1	2	5	4	NC	3	NC		**3,29**	1,4	0,52		0,1944
L222		NC	3	4	3	2	3	4	NC	4	NC		**3,29**	0,8	0,29		0,1944
L223		NC	NC	3	3	2	4	4	NC	3	NC		**3,17**	0,8	0,31		0,1873
L224		NC	NC	4	4	2	5	5	NC	4	NC		**4**	1,1	0,45		0,2366
L225		NC	NC	4	1	2	3	4	NC	5	NC		**3,17**	1,5	0,6		0,1873
Cost	**COSTES**	**2**	**2**	**3**	**2**	**NC**	**3**	**3**	**NC**	**3**	**1**		**2,38**	**0,7**	**0,26**		**0,1439**
Cost1		5	4	3	5	3	3	3	NC	3	5		**3,78**	1	0,32		0,2537
Cost2		5	4	3	5	3	3	4	NC	2	5		**3,78**	1,1	0,36		0,2537
Cost3		5	4	3	4	4	3	3	NC	3	4		**3,67**	0,7	0,24		0,2463
Cost4		5	3	3	5	4	3	3	NC	3	4		**3,67**	0,9	0,29		0,2463

Tabla III.1. Resultados de la encuesta de opinión sobre criterios de ontologías **(4/4)**.